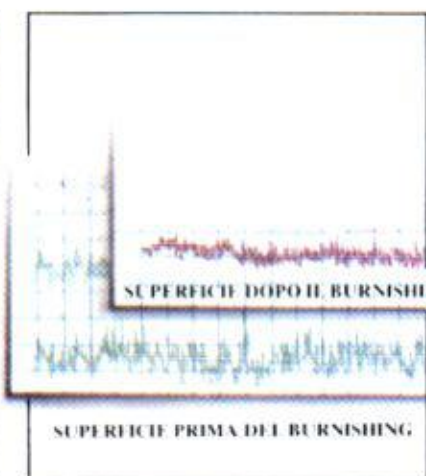
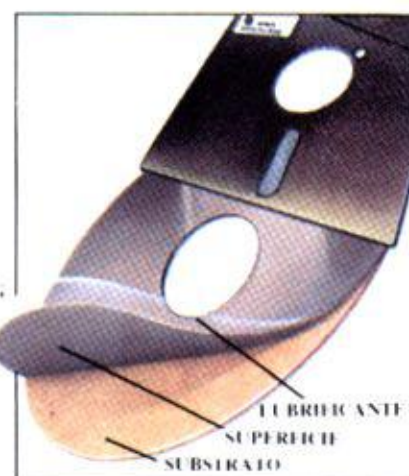
commodore
COMPUTER CLUB
16
L. 3.000
La rivista degli utenti di sistemi Commodore
Mensile - dicembre 1984 - Anno III - N. 16 - Sped. Abb. Post. Gr. III/70 - Distr. MePe
Utility:
Come vivere
di rendita
SALOON
HOTEL
PROGRAMMED BY ALLAN BLACK AND JON MALONE
GRAPHICS DESIGN BY MIKE MASTERS
COPYRIGHT(C)TALENT COMPUTER SYSTEMS 1984
Come creare
caratteri sul Vic 20
Nuovi comandi
Basic
Giochi per
C64 e Vic 20
systems

SCOPRI LA DIFFERENZA DYSAN

Perchè *Dysan*? Le Quattro Ragioni Per Preferire la Differenza Dysan

1. 100% di superficie testata "error free"

Solo Dysan garantisce che tutta la superficie della diskette sia realmente 100% "error free": un test esclusivo certifica le tracce e lo spazio tra le tracce assicurando prestazioni "error free" anche in presenza di disallineamento delle testine.

2. Esclusiva tecnica di Burnishing

Solo Dysan garantisce una superficie "a specchio" grazie alla sua avanzata ed unica tecnica di "burnishing" - questo risultato assicura un miglior segnale sulle tracce, una minor turbolenza sulle testine, consentendo un sicuro mantenimento dei dati dopo milioni e milioni di rotazioni.

3. Speciale lubrificazione

Solo Dysan garantisce, mediante uno speciale procedimento di lubrificazione, ottenuto trattando la superficie con il proprio esclusivo lubrificante DY 10, che le prestazioni "error free" siano esaltate e mantenute nel tempo.

4. Certificazione totale

Solo Dysan garantisce, con il suo metodo automatico di controllo qualità di tutta la produzione (risultato di una tecnologia leader nel mondo) che ogni diskette prodotta sia stata singolarmente testata e certificata.

Datamatic S.p.A.
via Volturno, 46
20124 Milano
tel.: 02/6073876 (5 linee r. a.)
telex: 315377 SADATA I

Filiale di Roma
via Città di Cascia, 29
tel. 06/3279987

Sommario

RUBRICHE

Direttore: Alessandro de Simone
Redazione/collaboratori: Giovanni Bellù, Andrea e Alberto Boriani, Giancarlo Castagna, Eugenio Coppari, Marco De Martino, Giancarlo Mariani, Enrico Scelsa, Fabio Sorgato, Danilo Toma
Segretaria di redazione: Maura Ceccaroli, Piera Perin
Impaginazione/illustrazioni: Francesco Amatori, Renato Caruso
Foto di copertina: Villa Iris S.n.c. - P.tta Massari 8 - Milano
Composizioni: Systems Editoriale S.r.l.
Fotolito: Systems Editoriale S.r.l.
Direzione, redazione: V.le Famagosta, 75 - 20142 Milano - Tel. 02/8467348
Pubblicità: Milano: Mirco Croce (coordinatore), Michela Prandini, Giorgio Ruffoni, Claudio Tidone, Villa Claudio - Segretaria: Liliana De Giorgi
• Roma: Spazio Nuovo - via P. Foscari 70 - 00139 Roma - Tel. 06/8209679
• Torino: Spaziobiù - via Filadelfia 50 - 10134 Torino - Tel 011/327617
Prezzo e abbonamento: prezzo per una copia L. 3.000. Arretrati il doppio. Abbonamento annuo (11 fascicoli) L. 28.000. Abbonamento annuo cumulativo alle riviste Computer e Commodore Computer Club (tariffa riservata agli studenti) L. 55.000. I versamenti vanno indirizzati a: Commodore Computer Club mediante assegno bancario o utilizzando il c/c postale n. 31532203
Stampa: La Litografica S.r.l. - Busto Arsizio
Registrazione: Tribunale di Milano n. 370 del 2/1/82 - Direttore Responsabile: Michele Di Pisa
Sped. in abb. post. gr. III - Pubblicità inferiore al 70%
Distribuzione: MePe, via G. Carcano 32 - Milano

Da questo mese la rivista C.C.C. è disponibile anche su nastro

Schede di espansione per Vic 20

☐ **Da quando ho acquistato una scheda di espansione da 16 K, i miei vecchi programmi non girano più. Come mai?**

● È molto probabile che i programmi cui ti riferisci siano dei giochini che utilizzano istruzioni del tipo POKE relative all'area di schermo. In questo caso non possono girare.

La memoria video del Vic, infatti, quando non è inserita alcuna espansione, occupa l'area che va da 7680 a 8186 (22 colonne × 23 righe) mentre l'area che contiene informazioni per i colori di ciascuna cella è quella numerata da 38400 a 38906.

Quando col Vic senza espansione digiti, ad esempio:

POKE 7680,1: POKE 38400,0 (R)

puoi notare che il carattere "A", in nero, compare in alto a sinistra sul video. Se pertanto, senza espansioni, realizzi un gioco che visualizza e colora caratteri sullo schermo mediante istruzioni POKE, gli argomenti di tali istruzioni saranno sicuramente compresi nell'area di memoria RAM prima detti.

Quando inserisci un'espansione RAM, i puntatori (cioè le informazioni da cui il Vic viene a sapere dove "si trova" lo schermo), vengono automaticamente alterati. Nel caso di espansioni da 8 oppure 16 K, i caratteri dello schermo, ed il loro colore, sono allocati, rispettivamente, da 4096 a 4602 e da 37888 a 38394.

Se carichi il programma cui accennavamo, questo eseguirà le POKE come prima, solo che ora, in quell'area, non c'è più la zona video.

In conclusione per utilizzare giochi nati per il Vic inespanso è necessario apportare le dovute modifiche oppure, ed è più semplice, rimuovere l'espansione prima di caricarli.

Listati di programmi protetti

☐ **È possibile avere i listati di programmi protetti? Molti di questi partono non appena termina il loro caricamento. Come individuare la SYS o l'indirizzo del RUN?**

● Molti programmi, oltre che protetti, sono scritti il più delle volte in Linguaggio Macchina e non in Basic. Alcuni, originariamente stesi in Basic, vengono in seguito compilati, sottoposti, cioè, ad un particolare procedimento che consente di velocizzare il programma.

Anche se riesci a rimuovere la protezione, è più che probabile che ti ritrovi un listato di alcune migliaia di istruzioni in L.M. che necessitano di un programma-utility, detto Disassembler, che "facilita" la loro interpretazione. Tale lavoro, però, richiede un'esperienza di programmazione consistente e non sempre è sufficiente. Se poi il programma non è stato originariamente scritto in L.M. ma è stato, da Basic, compilato, l'impresa è decisamente disperata.

In questi casi è molto più efficiente studiare ciò che riesce a fare il programma protetto e in seguito realizzare un listato in grado di fare le stesse cose.

Banche dati e trasmissioni via telefono

☐ **È possibile, con un Commodore 64, collegarsi ad una banca dati?**

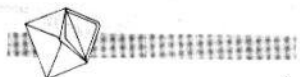
● Grazie ad un apparecchio chiamato MODEM, è possibile collegare, via telefono, qualsiasi computer con qualsiasi banca dati in grado di "dialogare" con esso. Il principio di funzionamento è piuttosto semplice ed è del tutto simile a quello che si realizza quando registri un programma col registratore.

Se, infatti, provi ad "ascoltare" un nastro su cui è inciso un listato, sentirai una serie di.. fischi che, per il computer, rappresentano ciò che per noi rappresenta la voce umana. Il programma, al momento della registrazione o della lettura, viene, in altre parole, trasformato in suono e, di conseguenza, è suscettibile di esser trattato come tale. È ovvio che il telefono, oltre che la tua voce, è un idoneo strumento per la "sua" voce. In genere due utenti, che possiedono ciascuno un computer, un modem ed un telefono, possono scambiarsi dati e programmi proprio ricorrendo al sistema descritto. È ovvio che dapprima uno dei due deve avvertire, telefonicamente, l'altro in modo che prepari il computer per ricevere i dati. Quando sono pronti per la rice-trasmissione inviano, premendo opportunamente dei tasti, i segnali.

Se però, ad inviare dati, è una grossa Società, tale sistema richiede parecchio tempo e, di conseguenza, la procedura viene automatizzata. In pratica un utente che desidera utilizzare una banca dati, paga un canone (fisso o proporzionale all'uso) alla Società. Questa gli invia, periodicamente, la parola d'ordine necessaria per entrare in collegamento con la banca. Come puoi immaginare un servizio di tal genere richiede, per essere conveniente (per la Società!), un gran numero di utenti e di denaro.

In America ne esistono parecchi, mentre in Italia esistono banche dati che vengono utilizzate da grosse industrie, ban-

che, società finanziarie capaci, comunque, di sostenere il costo, a volte notevole, del servizio.

Il successo di una banca dati è, tra l'altro, direttamente proporzionale al numero di apparecchi modem venduti. In Italia, purtroppo, siamo molto lontani dai livelli americani.

Le variabili del tempo

☐ Perché non è possibile modificare la variabile TI mentre ciò è possibile con TI$?

● Il motivo è piuttosto semplice ed indica, anzi, il buon criterio con cui sono stati realizzati i computer Commodore. Se l'utente, grazie ad una variabile, dispone già di un controllo sul tempo, a che serve permettergli di controllarne un'altra?

In altre parole è probabile che, durante l'elaborazione di un programma, l'utente modifichi più volte la variabile TI$ perdendo la... nozione del tempo. In qualsiasi momento, pertanto, un controllo su TI, inalterabile, permette di conoscere la quantità di tempo trascorsa da quando il computer è stato acceso oppure resettato via software.

È però possibile alterare il valore di TI agendo, mediante l'istruzione POKE, sulle tre locazioni che permettono al computer di calcolare TI. Le tre locazioni in questione sono 160, 161, 162 sia per il VIC 20 che per il Commodore 64.

Altera dapprima a piacimento il loro contenuto ed in seguito "chiedi" il valore di TI mediante un semplice PRINT TI.

L'Utility "Merge"

☐ In che consiste l'utility denominata "Merge"?

● Supponiamo che tu abbia registrato un programma Basic di 100 linee, numerate, però, da 1 a 50 e da 1000 a 1050.

Se possiedi un altro programma la cui numerazione sia compresa tra 51 a 999, l'utility Merge permette di fondere i due programmi in modo tale che la numerazione sia rispettata.

Dopo la fusione, in altre parole, nella memoria del calcolatore ottieni un unico programma Basic numerato correttamente. Questa routine è però di scarsa utilità perché capita piuttosto raramente avere un programma con un "buco" di numerazione in cui inserire un programma, guarda caso, che possiede proprio la numerazione mancante.

Programma caleidoscopio ed istruzione "Poke"

☐ Nonostante le precisazioni del N. 13 di C.C.C., il programma Caleidoscopio per il C-64 mi segnala Illegal Quantity Error in 260. Come mai?

● Il brevissimo programma in oggetto è stato pubblicato proprio per stimolare i nostri lettori. Ho fatto ricorso, infatti, ai simboli semigrafici invece del valore numerico ASCII (righe 150-170-180-190) in modo che il lettore scoprisse da solo quali fossero quei simboli sfruttando soprattutto le indicazioni di errore emesse dal computer. Il rigo 260 contiene, infatti, l'istruzione POKE che, se male utilizzata, emette il segnale di errore menzionato.

Esaminiamo, comunque, in dettaglio i casi critici.

L'istruzione POKE X, Y trascrive, nella locazione indicata con "X", il valore "Y". Due sono pertanto gli errori possibili.

Il primo consiste nel fornire un indirizzo "X" negativo o maggiore di 65536 (cioè l'ultimo byte della memoria, dato da 2 16).

Il secondo consiste nel fornire un valore negativo o maggiore di 255 (dato da 2 8).

Quando in una riga Basic è presente un'istruzione POKE X, Y e viene emessa la segnalazione di errore, questo non va ricercato nella linea indicata (come fareste per un Syntax Error) ma analizzando sia il valore "X" che "Y". Stabilito quale dei due non rispetta i limiti precedentemente accennati è necessario risalire alla linea responsabile dell'elaborazione di "X" ed "Y".

Nel caso particolare, hai sicuramente sbagliato nel trascrivere i caratteri semigrafici presenti sulle righe 180 e 190. Tale errata trascrizione rende negativi i valori di CL(5) o di CL(2) che, elaborati dalla linea 220, generano una "C" negativa che, a sua volta, rende impossibile l'esecuzione di riga 260. Quali sono, allora, i simboli giusti?

Scoprilo da te: ora devi riuscirci...

Programmi originali e non

☐ I programmi "Bioritmi" e "Archivio Dischi" pubblicati sui N. 4 e 10 della vo-

stra rivista sono stati tratti da precedenti pubblicazioni americane. Come conciliate questo fatto con l'affermazione che fate: affermate pubblicare solo listati originali?

● Il secondo programma cui ti riferisci rappresenta, in effetti, il rifacimento di un listato pubblicato su di una rivista americana di cui abbiamo i diritti di pubblicazione. C'è da dire che il listato originale era stato sviluppato per un doppio drive 4040 e, sul fascicolo di C.C.C. che citi sono presenti, invece, le modifiche che consentono di utilizzarlo anche sul singolo drive 1541.

Non abbiamo alcuna difficoltà ad affermare che abbiamo pubblicato altri listati, oltre a quello citato, che giravano sui "vecchi" PET Commodore e che rischiavano di essere trascurati perché incompatibili con il VIC 20 e Commodore 64.

Possiamo affermare, anzi, che vecchie versioni di listati interessanti saranno adattate ai computer in produzione e pubblicate in modo che i nuovi appassionati possano utilizzarli senza difficoltà. Il primo programma che citi, al contrario, era stato pubblicato supponendone l'originalità.

Ti ringraziamo, pertanto, della segnalazione e, anzi, allo scopo di evitare in futuro tali "incidenti" abbiamo deciso di regolarci nel modo seguente.

Il compenso stabilito per un articolo o programma verrà assegnato all'autore, (se effettivamente è... tale). In caso contrario verrà assegnato al lettore che per primo, con documentazione fotostatica, avrà fornito alla redazione la prova della... illegittima paternità del listato.

Le espansioni per il Vic 20

☐ **Mi capita di leggere, specie su riviste straniere, la pubblicità di alcune espansioni di memoria per il VIC 20 addirittura da 64 K RAM. Di che si tratta?**

● Anzitutto sconsigliamo di acquistare all'estero qualsiasi accessorio per computer. In Italia si ha, se non altro, la possibilità di... litigare in diretta in caso di malfunzionamenti.

Scherzi a parte le espansioni "ufficiali" per il VIC 20 sono quelle da 3K, 8K e 16K. È possibile, con un connettore particolare, ospitare più di un'espansione per volta.

Tieni presente, però, che l'espansione da 3K può coesistere insieme con quella da 8 oppure da 16K mentre una da 8 non può coesistere con quella da 16. Questo vale per le espansioni "ufficiali" (vendute, cioè dalla Commodore). Sono in commercio, però, alcune espansioni che consentono di riempire, da sole, tutta la RAM possibile che, oltre a quella fornita di serie col VIC 20, ammonta a 3 blocchi da 8K ed uno da 3K. Spieghiamoci meglio: il computer senza espansioni ha una quantità di RAM (gestibile da Basic) il cui indirizzo parte da 4096 e arriva a 7679. Aggiungendo la sola espansione di 3K il Basic viene automaticamente posizionato da 1024 a 7679.

Con l'espansione da 8K vengono utilizzati i byte da 4608 a 16383 mentre con quella da 16K si va da 4608 fino a 24575.

Il VIC 20, però, dispone di altri 8K di memoria indirizzabili vale a dire quelli da 24576 a 32767.

Gli indirizzi da 32768 in poi sono utilizzati dal computer per altri scopi e, di conseguenza, tali indirizzi non possono essere "riconosciuti" allo scopo di allo-

EDITORIALE · EDITORIALE

Ora che ce l'hai...

Molti di coloro che leggono queste note acquistano forse per la prima volta una rivista dedicata ai computer e tale ... evento è dovuto molto probabilmente al fatto che hanno appena acquistato un personal oppure stanno per acquistarlo.

L'anno scorso di verificò un Boom senza precedenti nella vendita dei calcolatori personali ed i costruttori, felicissimi, ne approfittarono per affermare che il computer stava finalmente prendendo il posto che gli spettava tra i regali di prestigio, intelligenti, "in", di moda ed altre cose del genere. Mentre, però, il numero di apparecchi venduti sale sempre più vertiginosamente, sarebbe necessario che qualcuno facesse, tra le tante, una statistica per controllare quanti acquirenti abbandonano il computer in cantina dopo i primi entusiasmi e le prime (più numerose) difficoltà.

Se chi legge queste righe è all'inizio, un solo consiglio: Datti da fare. Per gli altri, per i quali il computer è ormai pane quotidiano, non un consiglio ma un ordine: Impegnarsi maggiormente per restare un gradino più in alto rispetto agli altri.
E questo non per sterile antagonismo o competitività ma solo perché nel nostro futuro ci sarà sempre un computer.

Se ciò è un bene o un male non tocca a noi stabilirlo. Possiamo solo dire che, oggi più che mai, chi rimane fuori dal "giro" informatico sarà destinato a soccombere. Senza mezzi termini.

Alessandro de Simone

care programmi basic. In effetti c'è un altro blocco di memoria a disposizione ed è esattamente quello da 40960 a 49152 riservato alle cartucce dei giochi. Se in questi indirizzi poni una memoria RAM da 8K puoi disporre di memoria aggiuntiva che però, bada bene, non viene "vista" dal Basic ma che puoi utilizzare, ad esempio, per allocare programmi in linguaggio macchina.

Al di là di tali espansioni, e con le limitazioni dette, non è possibile gestire altra memoria.

Le espansioni da 64K cui ti riferisci sono in effetti dei veri e propri "doppioni" degli indirizzi del computer. Ciò significa che all'indirizzo 1000, tanto per dirne uno, sono presenti due locazioni RAM. L'una è quella "normale" gestita dal basic. L'altra, invece, è "nascosta" sotto di essa. Attraverso complessi programmi in linguaggio macchina è possibile selezionare la locazione di "sotto" oppure quella di "sopra" tenendo presente che con tali operazioni il sistema operativo (che si trova "sopra") può "scomparire" ed è necessario ricostruirlo in qualche modo. Tale tecnica viene adoperata normalmente nel Commodore 64 che, come è noto, è capace di gestire 64K di RAM proprio col sistema di selezione del banco di memoria che interessa.

In definitiva ti consigliamo l'acquisto di un banco da 64K per il VIC 20 solo se espertissimo in linguaggio macchina e se sei veramente innamorato del VIC, altrimenti ti consigliamo di lasciar perdere perché il gioco non vale la candela. Accontentati delle espansioni da inserire nelle locazioni lecite la cui quantità è del resto ragguardevole.

Alessandro de Simone

SI SCUSA...

Per motivi indipendenti dalla nostra volontà, una percentuale superiore alla media delle cassette di Commodore Club n. 2 presenta difficoltà di caricamento.

...E A DICEMBRE

RADDOPPIA !!!

Per evitare disagi ai nostri videolettori, Commodore Club n. 2 verrà duplicato a dicembre all'interno del terzo numero.
Per tutti, *quindi, Commodore Club n. 3 sarà un numero doppio!!!*
Tra gli altri meravigliosi giochi in edicola a dicembre:
Oroscopo, Pinocchio, Mr. Chomp, Shuttle, Visitors, Panettone...
e tanti altri.

I DISE

DI QUESTO FASC

In questo numero di C.C.C. il lettore troverà alcuni disegni (nove, per l'esattezza) realizzati con il plotter stampante Commodore a quattro colori. Tale scelta è stata fatta per rispondere, pur se parzialmente, alle numerose richieste di lettori, non proprio esperti, che dopo aver acquistato la periferica in oggetto son rimasti delusi per la mancanza di software ad essa dedicato. Come abbiamo più volte ripetuto, le schede pubblicate nell'ultima pagina della rivista sono tenute in grande considerazione e la pubblicazione delle routine ne sono una conferma.

Lo stesso autore dei listati, per non deludere i nostri lettori più fedeli che non possegono il plotter, ha sviluppato alcuni programmi che, utilizzando le routine grafiche di Danilo Toma (pubblicate sul N. 14 di C.C.C.) visualizzano sullo schermo gli stessi disegni a velocità, inutile dirlo, nettamente superiore.

Tali routine dimostrano, inequivocabilmente, che il programma di Toma pubblicato funziona perfettamente e NON CONTIENE ERRORI DI SORTA.

Altri listati che utilizzano le routine in oggetto stanno giungendo in Redazione e vengono attualmente analizzati per selezionarne i migliori.

Come realizzare i disegni pubblicati

I listati del primo gruppo (da utilizzare in unione col plotter), hanno tutti alcune linee in comune che, per non sottrarre spazio alla rivista, sono riportate una sola volta. La figura 1, infatti, contiene le linee che devono essere aggiunte a ciascun listato da utilizzare col plotter (figure 2, 3, ...10).

In ciascun listato la linea 120 indica il titolo del disegno mentre la linea 4 indica il tempo necessario per realizzare il disegno cui si riferisce la figura stessa.

I listati del secondo gruppo (figure 11, 12, ...17) sono invece "autosufficienti" e richiedono soltanto il preventivo caricamento (e lancio) delle routine di Toma pubblicate sul N. 14 di C.C.C.

Giancarlo Castagna

FIG. 1

```
1 REM **********************
2 REM * GIANCARLO CASTAGNA *
3 REM * ALCUNI DISEGNI PER *
5 REM * PLOTTER  COMMODORE *
6 REM **********************
10 :
15 REM LINEE COMUNI A TUTTE LE ROUTINE
16 REM PUBBLICATE IN QUESTO NUMERO
20 :
100 OPEN 1,6,1: OPEN 2,6,2
110 OPEN 4,6: OPEN 7,6,7
120 :
130 REM INSERIRE QUI LE ROUTINE
140 REM PUBBLICATE NEL FASCICOLO
240 :
250 PRINT#1, "H"
260 FOR I=1 TO 10:PRINT#4: NEXT
270 PRINT#1,"M",350,0: PRINT#4,"BY CAST"
280 FOR T=1 TO 4: PRINT#4: NEXT: PRINT#7
290 CLOSE 1: CLOSE 2: CLOSE 4: CLOSE 7
READY.
```

ICOLO

FIG. 3

```
4 REM * TEMPO : 4.00 MINUTI *
5 :
120 PRINT#4,"      *** QUADRATI COL. ***"
130 PRINT#1,"R",240,-240:PRINT#1,"I"
140 C=C+1:IFC>3THENC=1
145 PRINT#2,C
150 FORA=1.5TO.1STEP-.2
155 IFI=0+FTHENPRINT#1,"R",X,Y
160 FORI=0+FTO360+FSTEP90
170 II=(I/180)*π
180 X=A*(COS(II+(π/2)))*100
190 Y=A*(COS(II))*100
200 IFI=0+FTHENPRINT#1,"R",X,Y
210 PRINT#1,"J",X,Y
220 NEXTI:NEXTA
230 F=F+10:IFF=90THEN250
240 GOTO140
READY.
```

FIG. 4

```
4 REM * TEMPO : 5.45 MINUTI *
5 :
120 PRINT#4,"      *** STELLA E4 ***"
130 PRINT#1,"M",240,-240:PRINT#1,"I"
140 D=-1
150 FORA=.5TO1.5STEP.06
155 D=D+1:IFD=4THEND=0
156 PRINT#2,D
160 FORI=0TO360STEP22.5
170 C=C+1:B=A:II=(I/180)*π
180 IF(C/2)=INT(C/2)THENB=.5
190 X=B*COS(II+π/2)*100
200 Y=B*COS(II)*100
210 IFI=0THENPRINT#1,"R",X,Y
220 PRINT#1,"J",X,Y
230 NEXTI:NEXTA
235 Z=Z+1:IFZ<2THEN140
READY.
```

FIG. 2

```
4 REM * TEMPO : 6.30 MINUTI *
5 :
120 PRINT#4,"      *** QUADRATI ***"
130 PRINT#1,"R",240,-240:PRINT#1,"I"
140 PRINT#2,1
150 FORA=1.5TO.1STEP-.1
155 IFI=0+FTHENPRINT#1,"R",X,Y
160 FORI=0+FTO360+FSTEP90
170 II=(I/180)*π
180 X=A*(COS(II+(π/2)))*100
190 Y=A*(COS(II))*100
200 IFI=0+FTHENPRINT#1,"R",X,Y
210 PRINT#1,"J",X,Y
220 NEXTI:NEXTA
230 F=F+10:IFF=90THEN250
240 GOTO150
READY.
```

FIG. 5

```
4 FEM * TEMPO : 3.50 MINUTI *
5 :
120 PRINT#4,"      *** STELLA C1 ***"
130 PRINT#1,"M",240,-240:PRINT#1,"I"
140 D=-2
150 FORA=.5TO1.5STEP.05
155 D=D+2:IFD=4THEND=0
156 PRINT#2,D
160 FORI=0TO360STEP22.5
170 C=C+1:B=A:II=(I/180)*π
180 IF(C/2)=INT(C/2)THENB=A-.5
190 X=B*COS(II+π/2)*100
200 Y=B*COS(II)*100
210 IFI=0THENPRINT#1,"R",X,Y
220 PRINT#1,"J",X,Y
230 NEXTI:NEXTA
READY.
```

```
4 REM * TEMPO : 5.30 MINUTI *
5 :
120 PRINT#4,"     *** STELLA D4 ***"
130 PRINT#1,"M",240,-240:PRINT#1,"I"
140 D=-1
150 FORA=.5TO1.5STEP.05
155 D=D+1:IFD=4THEND=0
156 PRINT#2,D
160 FORI=0TO360STEP36
170 C=C+1:B=A:II=(I/180)*π
180 IF(C/2)=INT(C/2)THENB=A-.5
190 X=B*COS(II+π/2)*100
200 Y=B*COS(II)*100
210 IFI=0THENPRINT#1,"R",X,Y
220 PRINT#1,"J",X,Y
230 NEXTI:NEXTA
235 Z=Z+1:IFZ<2THEN140
READY.
```

FIG. 6

FIG. 8

```
4 REM * TEMPO : 5.30 MINUTI *
5 :
120 PRINT#4,"     *** PIANETA ***"
130 PRINT#1,"M",240,-240:PRINT#1,"I"
140 PRINT#2,2
150 FORA=0TO1.5STEP.1
160 FORI=0TO360STEP10
170 II=(I/180)*π
180 IFD=0THENB=1.5:C=A
190 IFD=1THENB=A:C=1.5
200 X=B*COS(II+π/2)*100
210 Y=C*COS(II)*100
220 IFI=0THENPRINT#1,"R",X,Y
230 PRINT#1,"J",X,Y
235 NEXTI:NEXTA
240 D=D+1:IFD<2THEN150
READY.
```

FIG. 9

```
4 REM * TEMPO : 8.30 MINUTI *
5 :
120 PRINT#4,"     *** STELLA F4 ***"
130 PRINT#1,"M",240,-240:PRINT#1,"I"
140 D=-1
150 FORA=.5TO1.5STEP.06
155 D=D+1:IFD=4THEND=0
156 PRINT#2,D
160 FORI=0+KTO360+KSTEP45
170 C=C+1:B=A:II=(I/180)*π
175 IFG<>0THENB=(A/2)+.25
180 IF(C/2)=INT(C/2)THENB=.25
190 X=B*COS(II+π/2)*100
200 Y=B*COS(II)*100
210 IFI=(0+K)THENPRINT#1,"R",X,Y
220 PRINT#1,"J",X,Y
230 NEXTI:NEXTA
235 Z=Z+1:IFZ<2THEN140
236 G=G+1:IFG<3THENK=22.5:GOTO140
READY.
```

```
4 REM * TEMPO : 5.30 MINUTI *
5 :
120 PRINT#4,"     *** CALEIDOSCOPIO ***"
130 PRINT#1,"M",240,-240:PRINT#1,"I"
140 D=-.25:DD=0
150 FORI=0TO360STEP2.5
160 D=D+.25:IFD=1THENDD=DD+1:D=0
165 IFDD=4THENDD=0
170 PRINT#2,DD
180 II=(I/180)*π
190 X=1.5*COS(II+π/2)*100
200 Y=1.5*COS(II)*100
210 PRINT#1,"H"
220 PRINT#1,"J",X,Y
230 NEXT
READY.
```

FIG. 7

```
4 REM * TEMPO : 10.30 MINUTI *
5 :
120 PRINT#4,"     *** PAVONE ***"
130 PRINT#1,"M",240,-240:PRINT#1,"I"
140 D=-1
150 FORI=0TO360STEP2.5
160 D=D+1:IFD=4THEND=0
170 PRINT#2,D
180 II=(I/180)*π
190 X=1.3*COS(II+π/4)*100
200 Y=1.3*COS(II)*100
210 PRINT#1,"H"
220 PRINT#1,"J",X,Y
230 NEXT
READY.
```

FIG. 10

COMMODORE 64

FIG. 11

```
4 REM *         QUADRATI         *
5 :
100 ←CLEAR:←GRAF1,6
150 L=4:FORB=π/18TO5*π/9STEPπ/18
200 FORA=10TO100STEP10
300 ←COLOR2:←ARC0,0,0,A,A,B,B+2*π,2*π/L
350 NEXTA:NEXTB
500 STOP
READY.
```

FIG. 12

```
4 REM *         STELLA E4         *
5 :
100 ←CLEAR:←GRAF1,6
150 FORA=25TO100STEP+10
160 I=0:D=0
170 IFE=1THENI=π/10
200 X=COS(I+π/2)*A
210 Y=COS(I)*A
215 I=I+2*π/16
220 X1=COS(I+π/2)*25
230 Y1=COS(I)*25
300 ←COLOR1:←DRAWX,Y,0,X1,Y1,0
350 I=I+2*π/16
360 X2=COS(I+π/2)*A
370 Y2=COS(I)*A
380 ←DRAWX1,Y1,0,X2,Y2,0
390 D=D+1:IFD<8THEN200
400 NEXT
410 E=E+1:IFE<2THEN150
500 STOP
READY.
```

FIG. 13

```
4 REM *         STELLA C1         *
5 :
100 ←CLEAR:←GRAF1,6
150 FORA=050TO100STEP+10
160 I=0:D=0
170 IFE=1THENI=π/10
200 X=COS(I+π/2)*A
210 Y=COS(I)*A
215 I=I+2*π/16
220 X1=COS(I+π/2)*(A-50)
230 Y1=COS(I)*(A-50)
300 ←COLOR1:←DRAWX,Y,0,X1,Y1,0
350 I=I+2*π/16
360 X2=COS(I+π/2)*A
370 Y2=COS(I)*A
380 ←DRAWX1,Y1,0,X2,Y2,0
390 D=D+1:IFD<8THEN200
400 NEXT
410 E=E+1:IFE<2THEN150
500 STOP
READY.
```

FIG. 14

```
4 REM *         STELLA D4         *
5 :
100 ←CLEAR:←MGRAF1,2,5,0
150 FORA=050TO100STEP+10
160 I=0:D=0
165 C=C+1:IFC=4THENC=1
170 IFE=1THENI=π/5
200 X=COS(I+π/2)*A
210 Y=COS(I)*A
215 I=I+2*π/10
220 X1=COS(I+π/2)*(A-50)
230 Y1=COS(I)*(A-50)
300 ←COLORC:←DRAWX,Y,0,X1,Y1,0
350 I=I+2*π/10
360 X2=COS(I+π/2)*A
370 Y2=COS(I)*A
380 ←DRAWX1,Y1,0,X2,Y2,0
390 D=D+1:IFD<5THEN200
400 NEXT
410 E=E+1:IFE<2THEN150
500 STOP
READY.
```

FIG. 15

```
4 REM *         PAVONE         *
5 :
100 ←CLEAR:←MGRAF1,6,5,0
170 FORI=0TO2*πSTEPπ/72
180 C=C+1:IFC=4THENC=1
200 X=COS(I+π/4)*100
210 Y=COS(I)*100
300 ←COLORC:←DRAW0,0,0,X,Y,0
400 NEXT
500 STOP
READY.
```

FIG. 16

```
4 REM *    CALEIDOSCOPIO    *
5 :
100 ←CLEAR:←MGRAF1,6,5,0
170 FORI=0TO2*πSTEPπ/72
180 C=C+1:IFC=4THENC=1
200 X=COS(I+π/2)*100
210 Y=COS(I)*100
300 ←COLORC:←DRAW0,0,0,X,Y,0
400 NEXT
500 STOP
READY.
```

FIG. 17

```
4 REM *         PIANETA         *
5 :
100 ←CLEAR:←GRAF1,5
200 L=40:FORA=00TO100  STEP+10
250 IFD=1THENB=A:C=100
260 IFD=0THENB=100:C=A
300 ←COLOR2:←ARC0,0,0,B,C,0,2*π,2*π/L
350 NEXTA
400 D=D+1:IFD<2THEN200
500 STOP
READY.
```

COMMODORE 64

Music 64
L. 255.000
+IVA

IL PREZZO COMPRENDE
TASTIERA, INTERFACCIA PER COMPUTER,
DUE PROGRAMMI
SU NASTRO E SU DISCO

Suonare uno strumento è bellissimo. Ma saper suonare tutti gli strumenti è proprio la fine del mondo. Scommettiamo che con il sistema Music 64 sei capace anche tu?

Se già possiedi un personal computer Commodore 64 con unità a cassette o a disco il gioco è fatto.

Ti basta acquistare, con una modica spesa, la tastiera, collegarla al computer e suonare! E se vuoi fare le cose alla grande puoi persino collegare il computer al tuo stereo.

Ma torniamo alla tastiera che, con ben 4 ottave, è davvero professionale. La tastiera ha due programmi. Il primo ti permette di sintetizzare suoni polifonici, così da potere eseguire accordi. Già predisposti ci sono i seguenti timbri: flauto, banjo, spinetta, campane tubolari e fisarmonica.

Il secondo programma è monofonico: premendo un tasto per volta puoi lanciarti in travolgenti assoli con il tuo strumento preferito: la riproduzione

ORCHESTRA

timbrica di ben 13 strumenti non ti lascia che l'imbarazzo della scelta.

Con un po' di pratica poi, riuscirai, intervenendo sui parametri, ad ottenere tutte le modificazioni di timbro che desideri.

E sempre modificando i parametri potrai persino ottenere effetti di vibrato, tremulo e modulazioni. E inoltre tutti gli effetti sonori più strabilianti e fantascientifici.

Tra breve Music 64 si arricchirà di nuove possibilità. Come la visualizzazione sul video del pentagramma relativo a ciò che stai suonando: e questo ti sarà di grande aiuto nello studio della musica.

Poi con Music 64 potrai memorizzare quello che suoni e intervenire su tutta la materia sonora programmata: per correggere, sviluppare o effettare a tuo piacimento.

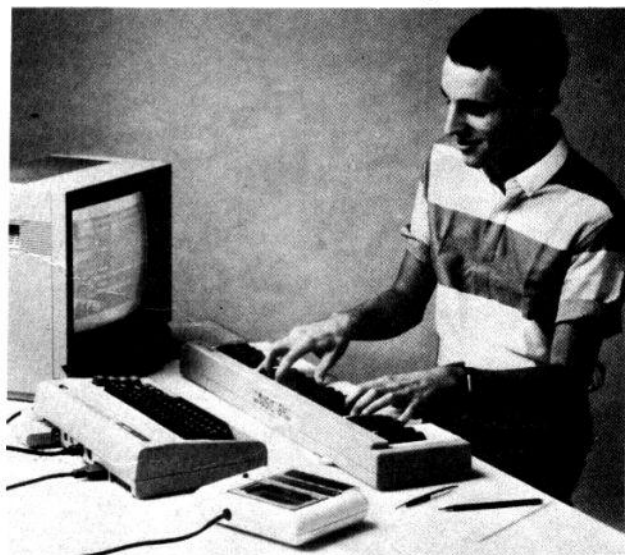
Ci sarà inoltre un programma di accompagnamento ritmico che prevede la possibilità di intervenire su 8 ritmi di base e di creare, a piacere, un nono ritmo del tutto nuovo. A te i tasti. E l'orchestra è tua.

Per ulteriori informazioni, compilare e spedire questo tagliando a:
SIRIUS ELETTRONICA s.r.l.
MILANOFIORI (ASSAGO) Palazzo F2 - 5° piano

Nome ______

Cognome ______

Via ______

CAP ______ Città ______

Music 64 si interfaccia con commodore

Distribuito da

sirius elettronica

MILANOFIORI (ASSAGO) Palazzo F2 - 5° piano - Tel. 8245321

IL MEGLIO DI COMMODORE COMPUTER CLUB

Siamo ormai giunti al numero 16 della rivista e molti degli argomenti affrontati sinora possono interessare i nuovi lettori della rivista.

Riportiamo qui di seguito i titoli, il numero del fascicolo ed un breve commento sul contenuto di articoli prevalentemente didattici.

- **Basic e linguaggio macchina** (n.1)
(Breve introduzione alle differenze tra i due linguaggi)
- **Simulazione di somma algebrica** (n. 3)
(Come il microprocessore esegue i calcoli)
- **L'aritmetica binaria** (n. 4)
(Considerazioni sul sistema di numerazione)
- **Simulazione monitor TIM** (n. 5 e 6)
(Semplice Monitor per chi inizia a lavorare in linguaggio macchina)
- **Assembler per tutti** (n. 6, 8, 13)
(Introduzione all'Assembler mediante semplici esempi sul CBM 3032, Commodore 64 e Vic 20)
- **Controllo del cursore mediante tastiera** (n. 7)
(Esempio pratico di comando del cursore in un gioco di facile comprensione)
- **Ed ora passo al Pet** (.7)
(Procedura da seguire per caricare sui PET serie 3000 - 4000 - 8000 programmi in Basic registrati con Vic 20 e Commodore 64)
- **I caratteri speciali dei Computer Commodore** (n.8)
(Quali sono, come ottenerli, come evitarli)
- **Il Buffet di Tastiera** (n. 8)
(Come utilizzare a proprio vantaggio alcune locazioni riservate al Sistema Operativo)
- **Le funzioni logiche** (n. 8)
(Simulazione dei circuiti logici ed elettronici presenti in un calcolatore: OR, AND, NAND, NOR, OR EX, NOR EX, INVERTER)
- **Tecniche di overlay** (n. 9)
(Come caricare in successione più programmi in modo che ognuno utilizzi i risultati delle elaborazioni precedenti)
- **Un pò d'ordine tra i bit** (n. 10)
(Come i calcolatori Commodore gestiscono la memoria per programmi, matrici, variabili)
- **Restore LN** (n. 10)
(Come limitare l'istruzione Restore solo ad alcune linee di DATA)
- **Routine grafiche il Linguaggio Macchina** (n. 10)
(Eccezionali routine per disegnare in alta risoluzione)
- **Come allungare le linee Basic** (n. 10)
(Basta con la limitazione di 80 caratteri per riga Basic)
- **Archivio dischi** (n. 10 e 11)
(Come realizzare un archivio dei propri dischetti allo scopo di rintracciare facilmente programmi e file)
- **La gestione dei file su cassetta** (n. 11)
(Come registrare dati su natro senza problemi)
- **Come usare correttamente joystick e paddle** (n. 12)
(Semplici esempi per predisporre il Vic 20 e il Commodore 64 agli utili accessori)
- **Alberi binari di ricerca** (n. 13)
(Come rintracciare dopo 16 tentativi un dato tra 65535)
- **Iper Monitor** (n. 13)
(Potente utility per lavorare in Linguaggio Macchina)
- **Disassembler** (n. 13)
(Utility per disassemblare listati in L.M.)
- **File Relativi** (n. 13)
(Aumentiamo la versatilità del drive 1541)
- **Espansione dei comandi Basic** (n. 13)
(Come aggiungere nuovi comandi al Commodore 64)
- **Interrupt** (n. 13)
(Come sfruttare una particolare routine dei computer Commodore per propri scopi)
- **Assembler** (n. 14)
(Brevissimo programma per lavorare in linguaggio Assembler)
- **Routine grafiche II** (n. 14)
(Nuovi potentissimi comandi Basic aggiuntivi per disegnare perfino in prospettiva)
- **Sette utility in Basic** (n. 15)
(Renumber, Append, Dump, Find, Delete, Gosub calcolato, Gosub Label)

COMMODORE 64 & VIC 20

Nuovi comandi BASIC

Come annunciato nel numero precedente, ecco un breve (rispetto al solito) programma che permette a chiunque di aggiungere nuovi comand al Basic.

Non è necessario sapere programmare in Linguaggio Macchina per usare questo programma: vi basta avere una routine in L.M., presa magari da una rivista (per esempio potete copiare quella proposta qui di seguito), metterla in memoria e fornire al mio programma l'indirizzo e il nome che intendete assegnarle. Lanciato con RUN 3000 il programma vi chiederà altri tre indirizzi... è tutto qui.

IL PROGRAMMA

Vediamo ora più in dettaglio le varie operazioni da compiere. Prima di tutto dovrete caricare le vostre routine ed allocarle in memoria, stando attenti che non vadano a sovrapporsi ad altre già presenti e che non si vengano a trovare in zone "pericolose", tipo la RAM accessibile ai programmi Basic o le locazioni a disposizione del registratore. Ricordatevi di prendere nota degli indirizzi di partenza di tali routines cioè, per intenderci, gli indirizzi che usate nelle SYS. Fatto ciò potete caricare questo programma. Prima di farlo girare dovrete scegliere le nuove "parole" e inserirle, insieme agli indirizzi delle routines a cui si riferiscono, in "speciali" linee Basic.

Speciali perché devono essere le prime linee del programma. Inoltre devono essere così strutturate "NUMERO DI LINEA" "PAROLA" = "INDIRIZZO". Nel listato vengono riportate, come esempio, le linee dalla 1000 alla 1800, che abilitano i vari comandi delle mie routines grafiche. D'ora in poi farò riferimento a questo tipo di linee chiamandole linee "comandi".

Due avvertimenti importanti: non lasciate spazi vuoti tra le parole (o tra le sue parti se è una parola composta) e il segno di uguaglianza. La routine del Sistema Operativo che prende i Bytes dal testo Basic scarta automaticamente tutti gli spazi e quindi un comando che ne contenga non verrebbe mai riconosciuto.

Secondo punto: i vostri nuovi comandi (le parole) vengono messi in una tabella che ha una lunghezza massima di 256 caratteri. Quindi il numero di comandi che potete aggiungere non è infinito (ma non è neppure troppo limitato). Ogni parola occupa tante locazioni della tabella quanti sono i caratteri che la compongono più una. Perciò ipotizzando una lunghezza media di cinque caratteri (sei locazioni nella tabella) potrete aggiungere INT (255/6) parole, cioè 42 comandi. Se superate il limite della tabella il programma si arresta e ve lo segnala.

L'ultima linea "comandi" deve essere seguita da una linea con un asterisco (il segno di moltiplicazione), per segnalare che le linee "comandi" sono terminate.

Battute le linee potete dare il RUN 3000.

Vi verranno chiesti tre indirizzi.

Per primo il programma vi chiederà dove volete allocare la routine (righe 24000-25000) che valuta i nuovi comandi e che è allocabile. Tale routine è identica a quela presente nelle mie routines grafiche (è infatti estratta da lì). A proposito, vi ricordo che all'interno di queste ultime viene collocata dalla locazione 51000 alla 51115 (linee di DATA dalla 2240 alla 2330 del programma del N° 14 di C.C.C.).

Valgono anche in questo caso e nei due seguenti le raccomandazioni di prudenza per non allocare routine e tabelle in posti "sbagliati". Quando avrete fornito l'indirizzo e premuto RETURN verranno letti i DATA (cioè la routine) e posti in memoria. Vi ricordo ancora una volta che è importantissimo trascrivere esattamente i DATA, pena il malfunzionamento o l'inchiodamento della macchina. Per aiutarvi nel controllo ho inserito una linea che arresta il programma se la somma dei DATA non è quella giusta. Come già spiegato nel numero precedente non è un sistema infallibile, per cui salvate il programma prima di abilitare la nuova routine di valutazione

comandi. La seconda cosa che vi verrà chiesta è dove volete collocare la tabella delle parole. Nelle routine grafiche del N° 14 si trova dalla locazione 51116 alla locazione 51163.

Badate che ogni volta che usate questo programma le tabelle vengono riscritte utilizzando soltanto parole e indirizzi presenti nelle linee "comandi": quindi dovete "dichiarare" ogni volta tutti i comandi che volete siano riconosciuti. Per ultimo vi verrà chiesto di indicare dove volete collocare la tabella degli indirizzi delle routines che eseguono i nuovi comandi. Nelle già citate routines grafiche questa tabella viene posta nelle locazioni 50172-50189, sfruttando uno "spazio inutilizzato" che si estende fino alla locazione 50214. Se intendete utilizzare tali routines tenete conto che in questo spazio potrete collocare tabelle lunghe al massimo 43 Bytes. Ogni indirizzo della tabella "indirizzi" occupa due locazioni ed anche qui c'è il limite dei 256 Bytes, quindi il numero massimo di indirizzi inseribile è 128. Finalmente, dopo avere ancora una volta premuto RETURN, verranno compilate le tabelle e visualizzate parole e indirizzi inseriti. Per ultimi verranno scritti il numero dei nuovi comandi e i Bytes occupati dalle due tabelle.

Prima di abilitare i nuovi comandi verrà chiesto se tutto è in ordine. Se, pertanto, avete dubbi su nomi o indirizzi (sia delle routines che delle tabelle), rispondete "NO" ed il programma si fermerà, consentendo di effettuare le opportune modifiche.

Rispondendo "SI", verrà eseguito il comando SYS (R) (riga 6450) in cui "R" e l'indirizzo che avete fornito rispondendo alla prima richiesta (allocazione routine nuovi comandi). Da quel momento i nuovi comandi saranno attivati.

Alcune parole visualizzate potranno apparire parzialmente diverse da quelle da voi scelte, in particolare con un segno grafico al posto di due o più lettere. Non preoccupatevi, è tutto a posto, solo che nella vostra parola è presente una parola del "vecchio" Basic e la sua rappresentazione in memoria differisce da quella che appare nel listato.

Chi ha presente come funziona l'interprete Basic, avrà già capito che la responsabile di tutto ciò è la routine del Sistema Operativo che compatta (o "tokenizza") i comandi prima di eseguirli (se sono diretti) o di metterli in memoria (se si trovano in linee di programma). Tale routine riconosce le vecchi parole e le sostituisce con un numero di codice anche se sono "nascoste" dentro ad altre, tranne nel caso siano su una linea di DATA o siano precedute dalle virgolette (quelle usate nel comando PRINT), anche nelle REM.

È per tale motivo che non ho potuto utilizzare linee di DATA ma ho dovuto fare ricorso allo strano metodo delle linee "comando" per compilare le tabelle. La rappresentazine della parola viene letta direttamente in memoria e posta nella tabella senza più rischi di una errata interpretazione.

A proposito di interpretazioni, devo avvertirvi che se vi capita di scegliere una parola (es. PINO) UGUALE ALLA PRIMA PARTE di un'altra (es. PINOCCHIO), dovete porla in linea "comando" SUCCESSIVA a quest'ultima. Perciò, ad esempio, sarà corretto così:

100 PINOCCHIO = xxxxx
200 PINO = yyyyy

Se farete il contrario, cioè definirete prima la parola più corta, non riuscirete a fare eseguire il comando dalla parola più lunga (PINOCCHIO).

ELIMINAZIONE DELLA FRECCETTA

Fino ad ora non ve l'ho detto ma, visti i riferimenti alle routines grafiche del numero scorso, avrete già intuito che i nuovi comandi per essere riconosciuti devono essere preceduti da una freccetta a sinistra: "←". Come promesso però ecco qui la formula magica per eliminare la poco ortodossa freccia.

Sostituire il settimo DATA della linea 24300 che è 121 con 115 (la somma dei DATA di riga 3025 di conseguenza diventa 16076). Nella linea 3060 invece sostituite i due 11 con due 24. È tutto qui.

Chi vuole, può addirittura cambiare direttamente i DATA delle routines grafiche senza ricorrere a questo programma: cambiate il primo DATA della linea 24250 (linea del programma "routines grafiche II") da 67 in 80 e il settimo DATA della linea 24300 da 121 in 115. Quindi l'ultima somma di controllo di quel programma va cambiata. Da 18956 a 18963 (linea 17).

L'eliminazine della freccetta comporta un piccolo svantaggio e cioè un certo rallentamento nell'esecuzione dei "vecchi" comandi Basic. Vediamo come. Quando viene dato un comando o fatto girare un programma entra in azione la mia routine che controlla se il comando da eseguire è preceduto da una freccia e se ciò non è cede il controllo alla routine "originale" e la perdita di tempo è di pochi microsecondi.

Se invece viene eliminata (nel modo indicato) la freccia di "riconoscimento", la mia routine confronterà il comando con tutte le parole della nuova tabella e solo allora cederà il controllo al "vecchio" Basic. Il ritardo diventa proporzionale al numero e alla lunghezza dei nuovi comandi.

Mi sembra di avere spiegato tutto perciò ora non vi resta che provare ad utilizzare il programma.

Per non correre rischi di inchiodamento, assicuratevi che, quando digitate "RUN 3000" la routine che valuta i comandi sia quella originale. Il modo più sicuro è ricorrere a SYS 64738, che però vi fa perdere il programma in memoria. Quindi ricordatevi di farlo prima di caricarlo.

TRASFORMARE IN DATA, O...

Certo eseguire queste operazioni ogni volta che si accende il computer è disagevole perciò quando avrete messo a punto le vostre routines, le avrete caricate in memoria e avrete fatto girare questo programma vi converrà "fissare" tutto in un unico programma. Per fare ciò avete due possibilità. Un primo modo è quello di trasferire routines e tabelle in un unico blocco di DATA, come ho fatto per le routines proposte nel N. 14.

L'operazione non presenta grosse difficoltà ma, specialmente se i DATA sono 2000, può risultare veramente pesante. Per questo mi sono fatto un piccolo programma che legge il contenuto della memoria e lo trasforma in linee di DATA. Se vi interessa telefonate alla redazione (non a me per favore). Se ci saranno abbastanza richieste lo vedrete presto su queste pagine. Un secondo e più veloce modo per "saldare i vostri nuovi comandi è di salvare direttamente la parte di memoria che contiene routines e tabelle.

Il modo per farlo è abbastanza semplice ed è il mio preferito. Quando date il comando SAVE la macchina prende le informazioni su quale parte della memoria salvare da due puntatori: il primo (che indica dove iniziano i programmi Basic), situato nelle locazioni 43 e 44, stabilisce la prima locazione da salvare, il secondo (fine programma Basic), situato nelle locazioni 45 e 46, stabilisce l'ultima.

Basta dunque modificare questi puntatori e potremo salvare le nostre routines. Per sapere quali valori pokare in queste quattro locazioni fate in questo modo. Poniamo che X sia l'indirizzo iniziale.

```
HI = INT (X/256) : LO = X-HI*256
POKE 43, LO : POKE 44, HI
```

Gli stessi calcoli dovranno essere fatti per l'indirizzo finale ma le POKE saranno dirette alle locazioni 45 e 46.

Fatto ciò date il SAVE. Ricordate, dopo avere effettuato la registrazione, di rimettere i valori "normali", che avrete avuto cura di segnarvi, nei due puntatori oppure ricorrete al solito SYS 64738.

Quando caricate le routines così salvate dovete aggiungere un paio di "cose" al comando LOAD. Dovrete infatti digitare:

LOAD "nome programma", n. 1

Dove n. è il numero di periferica, che sarà 1 per il registratore e 8 per il disco. Se non ricordate il nome del programma e state usando il registratore potete battere le virgolette senza interporre spazi (LOAD "", 1,1).

Il comando LOAD, se non viene specificato nulla oltre al nome del programma da caricare, colloca tale programma invariabilmente a partire dalla locazione 2049 (inizio area programmi Basici mentre se gli si dà come indirizzo secondario "1" lo colloca nella stessa posizione in cui si trovava quando è stato salvato.

Quindi una volta finito il caricamento le routines e le tabelle saranno già in "posizione" e pronte ad essere attivate con una SYS. Il risparmio è doppio perché vi risparmiate il tempo necessario alla lettura dei DATA ed inoltre la registrazione "diretta" anziché attraverso linee di DATA occupa meno spazio sul nastro (e sul disco) e quindi il caricamento dura meno tempo.

Dimenticavo di dirvi che usando il comando LOAD in questo modo dovrete rimettere a posto i soliti puntatori. Ricorrete ancora una volta a SYS 64738.

Tutta la faccenda vi potrà apparire inizialmente complicata ma in realtà è molto semplice e rapida: accendete il computer, caricate i vostri nuovi comandi nel modo indicato, SYS 64738 e SYS X in cui "X" é l'indirizzo di partenza della nuova routine di valutazione comandi.

UNA APPLICAZIONE

La breve routine qui di seguito riportata aggiunge un nuovo comando alle routines grafiche.

Questi pochi ma utili DATA permettono di trasferire la pagina grafica in una zona della memoria accessibile al comando PEEK. Dalla locazione 32768 alla loc. 40767. Potrete così, con opportuni programmi, salvare i vostri disegni oppure farli stampare.

L'operazione contraria (cioè il trasferimento dalla memoria di massa alla pagina grafica) non necessita di questo passaggio intermedio (il comando POKE funziona anche nella pagina grafica) per cui non ho ritenuto il caso di "sprecare" altri DATA.

Ricordate che questa routine funziona solo se sono presenti le routines grafiche. Da sola porta ad un quasi sicuro inchiodamento.

Procedete come spiegato e avrete l'immensa gioia di vedere che il vostro 64 non vi risponderà con un perentorio SYNTAX ERROR quando gli direte di eseguire il vostro nuovo comando.

UN ESEMPIO

Descrivo ora il procedimento da seguire nell'ipotesi che il lettore desideri utilizzare una routine in L.M. non servendosi di un comando SYS ma di un comando Basic di sua... creazione. Supponiamo che la routine in oggetto sia quella del programma di figura 2. Questa consente di cambiare il colore del bordo e dello sfondo dello schermo. È ovvio che la sua banalità serve solo da esempio per meglio comprendere la procedura da seguire per aggiungere nuove istruzioni Basic.

● Caricate il programma di figura 2 e fatelo partire con RUN.

● Rispondete con "30000" alla domanda "LOCAZIONE INIZIALE?". In questo modo la routine in L.M. (contenuta nelle righe DATA) sarà trascritta a partire dalla locazione 30000 in poi. Attenzione: la routine di figura 2 è interamente rilocabile e, di conseguenza, "funziona" in qualsiasi parte della memoria RAM la allocate. Per ciò che riguarda altre routine in L.M. è probabile che esse girino solo se allocate a partire da una ben determinata locazione. In questi casi è INDISPENSABILE rispettare il giusto indirizzo altrimenti il 64 rischia l'inchiodamento.

● Caricate il programma di figura 1. In questo (riga 2800) potete notare che ho già predisposto la riga-comando:

2800 SCHERMO = 30000

La parola codice "SCHERMO" potete modificarla, se credete.

● Digitate RUN 3000. La domanda "ALLOCAZIONE ROUTINE?" si riferisce, come già detto, NON alla routine in L.M. del programma prima caricato (di figura 2) ma al programma in Linguaggio Macchina (anch'esso interamente rilocabile) che è posto nei DATA da riga 24250 a riga 25000. Questa interessante opportunità vi consente di "piazzare" dove meglio credete la mia routine lasciandovi ampia libertà per allocare le vostre. Digitate, per esempio, 25000 alla domanda in oggetto e 26000 e 27000 alle altre due domande che seguono.

● Digitate SI (quando siete sicuri di ciò che avete fatto) alla domanda "POSSO ATTIVARE I NUOVI COMANDI?"

● Da questo momento il Commodore 64 accetterà il nuovo comando:

SCHERMO A,B

in cui "A" rappresenta il colore dello sfondo e "B" quello del bordo.

NOTA BENE

La nuova routine di valutazione comandi si occupa di riconoscere la "parola" del comando e non eventuali parametri che la seguono. Questo compito spetta alle routine di "servizio", cioè a quelle che eseguono i comandi (in questo caso ai 27 bytes della microroutine applicativa).

Se conoscete il linguaggio macchina ma non conoscete il modo di operare dell'interprete Basic, dovrete accontentarvi di creare nuovi comandi esclusivamente senza parametri che li seguano (un esempio di comandi senza parametri è "CLEAR" delle routine grafiche del N. 14).

Naturalmente potrete sempre ovviare al bisogno di "passare" delle informazioni delle routines di "servizio" utilizzando le famigerate POKE.

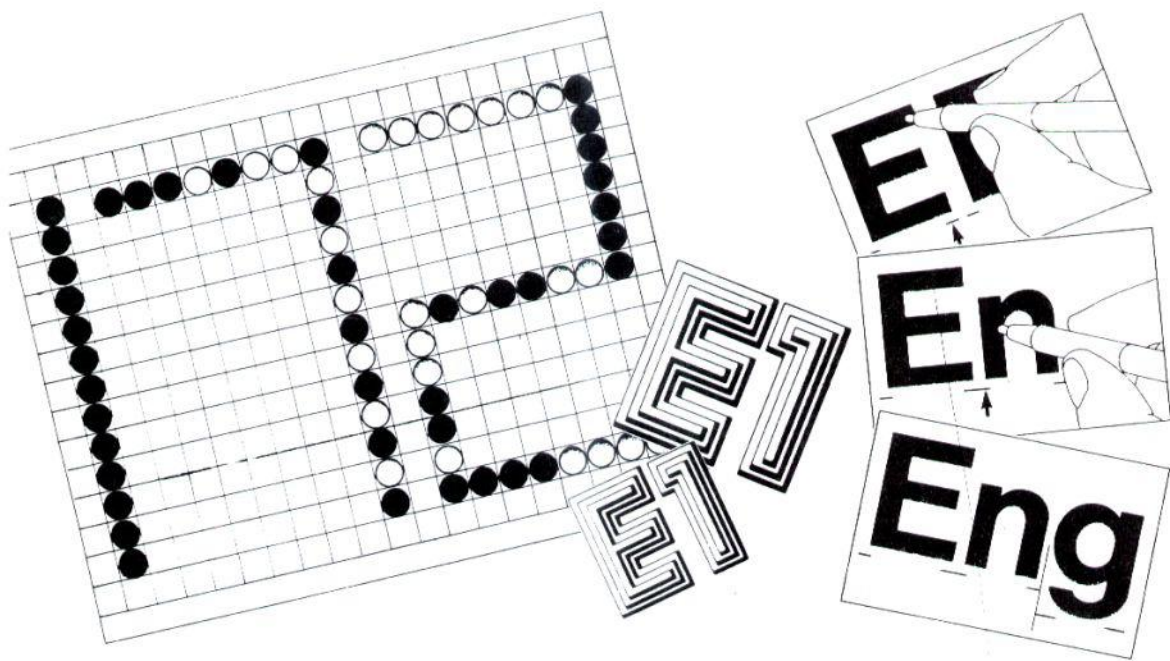
ALTRA NOTA

Se farete girare la microroutine applicativa con una SYS otterrete solo un SYNTAX ERROR. Questo perché, come avrete già capito, la microroutine cerca i parametri della sua "istruzione madre" e, non trovandola.. SYNTAX ERROR. Infine una dolorosa scoperta dell'utimo momento: esiste un caso in cui i nuovi comandi non vengono riconosciuti ed il computer si vendica con una segnalazione di errore. Ciò succede quando il nuovo comando si trova subito dopo l'istruzione THEN. Per ovviare all'inconveniente basta inserire, tra THEN ed il nuovo comando il carattere di doppio punto (:).

Danilo Toma

```
10 REM *** ROUTINE PER SETTARE IL COLORE ***
20 REM ***    DELLO SFONDO E DEL BORDO    ***
30 :
40 REM ***   VA USATA INSIEME ALLA NUOVA ***
50 REM ***   ROUTINE DI VALUTAZ. COMANDI ***
90 :
100 INPUT"[illegible]LOCAZIONE INIZIALE";Y
150 FORI=YTOY+26:READA:POKEI,A:B=B+A:NEXT
200 IFB<>3179THENPRINT"ERRORE NEI DATA"
999 :
11100 DATA 32,158,183,224,16,48,3
11200 DATA 76,72,178,142,33,208
11300 DATA 32,253,174,32,158,183,224
11400 DATA 16,16,240,142,32,208,96
READY.


2800 SCHERMO=30000
2898 *
2899 :
2900 REM ******************************
2901 REM ***       COMMODORE   64      ***
2905 REM ***  ROUTINE NUOVI COMANDI    ***
2910 REM ***        DANILO TOMA        ***
```

```
2924 REM ***   (BATTERE: RUN 3000)    ***
2925 REM ******************************
2999 :
3000 L$="INDIRIZZO":T$=" TABELLA"
3010 PRINT"■■"L$" ALLOCAZ. ROUTINE";:INPUTR
3020 FORI=RTOR+115:READA:POKEI,A:B=B+A:NEXT
3025 IFB<>16082THENPRINT"ERRORE NEI DATA":END
3030 PRINTL$;T$" NUOVE PAROLE";:INPUTI:JI=I
3040 PRINTL$;T$" INDIRIZZI";:INPUTK:PRINT
3050 M=PEEK(43)+PEEK(44)*256-1:S=R+51
3060 HI=INT((R+11)/256):LO=R+11-HI*256:POKER+1,LO:POKER+6,HI
3070 HI=INT(I/256):LO=I-HI*256:POKES+10,LO
3080 POKES+11,HI:POKES+20,LO:POKES+21,HI
3100 HI=INT(K/256):LO=K-HI*256:POKES+34,LO
3110 POKES+35,HI:POKES+41,LO:POKES+42,HI
3120 M=M+5:J=I+1:PRINT,
3150 T=PEEK(M):IFT=172THENPOKES,C:GOTO6000
3200 IFT=178THEN3500
3220 IFJ-JI<256GOTO3260
3240 PRINT:PRINT"HAI SUPERATO IL LIMITE DELLA "T$:END
3260 IFT=0THENPRINT:PRINT"MANCA INDIRIZZO":END
3300 IFT<128THENPRINTCHR$(T);
3350 IFT>127THENPRINTCHR$(97);
3400 POKEJ,T:J=J+1:M=M+1:GOTO3150
3500 POKEI,J-I-1:I=J
3900 R$="":M=M+1:T=PEEK(M):IFT=0THEN3250
4000 T=PEEK(M):IFT=0THEN4500
4100 R$=R$+CHR$(T):M=M+1:GOTO4000
4500 T=VAL(R$):IFT<0ORT>65536THENPRINT:PRINT"INDIRIZZO ILLEGALE":END
4600 HI=INT(T/256):LO=T-HI*256
4700 PRINTHI*256+LO:POKEK,LO:POKEK+1,HI
4800 K=K+2:C=C+1:GOTO3120
6000 PRINT:PRINT" COMANDI";C:PRINTT$" PAROLE";J-1-JI"BYTES"
6100 PRINTT$" INDIRIZZI"2*C;"BYTES"
6200 INPUT"■ POSSO ATTIVARE I NUOVI COMANDI";KK$
6300 IFKK$="NO"THENEND
6400 IFKK$<>"SI"THEN6200
6450 SYSR: REM *** ABILITA NUOVA ROUTINE VALUTAZIONE COMANDI ***
6500 PRINT"■■"TAB(15)"T.D. BASIC"
7000 END
9999 :
24000 REM ***  ROUTINE IN LINGUAGGIO MACCHINA  ***
24250 DATA 169,67,141,8,3,169,199,141,9,3,96,32,115
24300 DATA 0,201,95,240,6,32,121,0,76,231,167,169,0
24400 DATA 133,249,165,122,133,252,165,123,133,253,160,0,240
24500 DATA 20,165,252,133,122,165,253,133,123,230,249,169,9
24600 DATA 197,249,240,218,200,202,208,252,185,172,199,170,32
24700 DATA 115,0,133,254,200,185,172,199,197,254,208,219,202
24800 DATA 208,240,6,249,166,249,189,252,195,141,63,3,232
24900 DATA 189,252.195,141,64,3,169,32,141,62,3,169,96
25000 DATA 141,65,3,32,115,0,32,62,3,76,174,167
READY.
```

Creazione di Caratteri sul Vic 20

Il VIC 20, come il fratello maggiore CBM 64, è stato dotato da Mamma Commodore di un ottimo set di caratteri: maiuscoli, minuscoli, grafici e speciali. Se vogliamo creare un video-vic-gioco non possiamo usare tali caratteri ma dobbiamo poter disegnare mostri intergalattici o cose simili. Come possiamo farlo se i caratteri del VIC si trovano in una ROM, dove possiamo solo andare a leggere (Read Only Memory)? Vediamo come è possibile farlo.

UN PO' DI TEORIA

Il nome che il nostro "cervellone" ha, mi riferisco al VIC 20, è: VIC V-video I-nterface C-hip. In pratica un integrato destinato alla gestione del video. Molte delle funzioni del VIC 20 sono così da attribuire a questo integrato denominato con la sigla 6561.

A tale Chip sono state riservate tre aree di memoria:

- Mappa Video in Ram
- Mappa Colori in Ram
- Mappa Caratteri in Rom

L'indirizzo di tali aree dipende dalla configurazione di memoria in cui ci troviamo, esclusa la Mappa Caratteri che inizia sempre a 32768-$8000.

Più precisamente avremo:

M. Video: ● 7680 - ($1 e 00) a 8191 - ($1fff) senza espansioni o con +3k
● 4096 - ($1000) a 7679 - ($1dff) con espansioni a partire da +8k

M. Colori: ● 38400 - ($9600) a 38911 - ($97ff) senza espansioni o +3k
● 37888 - ($9400) a 38399 - ($95ff) con espansioni maggiori a +3k

Veniamo ora al punto cruciale della situazione. Il Chip 6561 possiede al suo interno ben 16 registri di memoria a 8 bit ciascuno: 36864 - ($9000) a 36879 - ($900f). Tali registri sono ovviamente posti in Ram, e quindi è possibile programmarli dall'utente.

All'accensione della macchina vengono inizializzati dal sistema in uno schema standard: video da 23 righe per 24 colonne, schermo bianco e bordo ciano, caratteri bleu, e quello che più ci interessa generatore di caratteri con inizio a 32768 - ($8000).

Dopo tutti questi discorsi, avrete sicuramente intuito dove voglio arrivare: l'in-

dirizzo di inizio del generatore di caratteri o Mappa Caratteri è accessibile via software dall'utente. Più precisamente, vi sarà uno dei 16 registri del 6561 adibito a questo. Infatti è proprio così.

Il registro in questione è il byte di indirizzo 36869 - ($9005) o, per meglio dire, sono i bit 0-3 di tale registro che indicano l'inizio della Mappa Caratteri. Comprendere come questi 4 bit vengano utilizzati dal sistema operativo per determinare l'indirizzo del generatore caratteri, richiederebbe una conoscenza approfondita dell'hardware della macchina, e quindi sorvoliamo. Vediamo invece gli altri registri, che possono tornare utili per determinare applicazioni, soprattutto grafiche.

Come già detto a noi interessa soprattutto il registro #6, o meglio interessano solo i bit 0-3 di tale registro. Il bit 3, se resettato, cioè posto a 0, dice al 'cervellone' di andare a pescare nella Rom generatrice di caratteri a partire dall'indirizzo dato dai tre bit 0-2 (da notare il fatto che l'indirizzo di inizio della mappa caratteri, sia in Rom che in Ram, sarà sempre all'inizio di un blocco da 1k). Se invece il bit 3 è settato, cioè posto a 1, il calcolatore andrà a pescare nella Ram.

I possibili indirizzi utilizzabili in Ram sono i seguenti:

BIT	3	2	1	0	DEC	ESA
	1	1	0	0	4096-	$1000
	1	1	0	1	5120-	$1400
	1	1	1	0	6144-	$1800
	1	1	1	1	7168-	$1c00

È ovvio che per sceglierne uno dovremo tener conto della configurazione di memoria in cui ci troviamo, e della lunghezza del programma che utilizza la nuova Mappa (es. non possiamo porre il generatore di caratteri a 4096 se abbiamo un Vic 20 in versione base, infatti il Basic con questa versione del Vic inizia proprio a tale indirizzo). Potremmo porre la Mappa caratteri con un Vic Base a 5120, 6144, 7168 ed avere così rispettivamente una Mappa caratteri da 3k, 2k, 1k.

Registri 6561

N.	DEC	ESA	
#1 -	36864	$9000	bit 0-6 distanza prima colonna dal bordo sinistro
#2 -	36865	$9001	bit 0-7 distanza prima riga dal bordo superiore
#3 -	36866	$9002	bit 0-6 numero colonne
			bit 7 fa parte dell'indirizzo d'inizio Mappa Video
#4 -	36867	$9003	bit 1-6 numero righe
			bit 0 seleziona il formato dei caratteri 0=8*8, 1=16*8
#5 -	36868	$9004	bit 0-7 posizione numero linea raggio di scansione video
#6 -	36869	$9005	bit 0-3 inizio generatore caratteri
			bit 4-7 indirizzo Mappa Video

Utilizzando una espansione da +3k dove il basic inizia a 1025- ($0401) potremo porla anche a 4096, e non più "bassa" perché il 6561 non può raggiungere indirizzi di memoria minori di 4096.

Mentre, se disponiamo di una espansione da +8k, +16k possiamo porla solamente a 6144 e 7168. In questo ultimo caso possiamo utilizzare un piccolo trucco e porre la Mappa caratteri prima del basic, basta cioè spostare l'inizio del basic in avanti e riservarci la Ram che vogliamo.

PASSIAMO ALLA PRATICA

Vediamo ora come utilizzare quello che abbiamo imparato sul 6561. Coma abbiamo visto, il registro da manipolare è il #6. La regola per calcolare che valore bisogna "pokare" in tale registro per un dato indirizzo può essere la seguente:

Poke 36869 (Peek 36869) and MC) OR (SA/1024+8)

con: ● mc × = 240 con Vic base o +3k × = 192 con espansioni maggiori a +3k
● sa = indirizzo di inizio per nuova mappa caratteri.

Quindi, se ad esempio vogliamo porre la mappa caratteri a 5120, con un Vic in versione base, scriveremo:

Poke 36869, ((Peek (36869) and 240) or 13)

che corrisponde a: Poke 36869, 253

È comunque consigliabile, quando si scrivono programmi, di utilizzare la regola generale per poter in seguito adattare il programma ad ogni configurazione di memoria senza dover stare a correggere tutto il listato.

Come esempio di quanto detto sopra provate a dare il seguente comando (quanto segue è valido per un Vic con espansione non maggiore ai 3k):

Poke 36869,242

Vedrete i caratteri trasformarsi in minuscolo (è la stessa situazione in cui vi trovate se premete contemporaneamente i tasti shit+ commodore): avete obbligato il Vic ad andare a leggere i caratteri nella seconda parte del generatore, appunto quella che contiene il minuscolo.

Per tornare nella situazione normale basta scrivere:

Poke 36869,240

ovviamente seguito dal tasto 'return'.

Che dobbiamo fare, in pratica, se vogliamo disegnare il nostro mostro intergalattico sul Vic? Basterà trasportare il generatore di caratteri in Ram e sostituire ad un carattere standard, ad esempio la d, il nostro mostro.

Innanzitutto vediamo come sono definiti i caratteri nel Vic 20. Un carattere è costituito da 8 byte consecutivi, è quindi formato da 64 punti (8*8 = 64). Ad ogni punto acceso sul video corrisponde un 1, mentre ad ogni punto spento corrisponde uno zero.

Quindi la lettera 'A' sarà così formata:

	7	6	5	4	3	2	1	0
0	0	0	0	1	1	0	0	0
1	0	0	1	0	0	1	0	0
2	0	1	0	0	0	0	1	0
3	0	1	1	1	1	1	1	0
4	0	1	0	0	0	0	1	0

5	0	1	0	0	0	0	1	0
6	0	1	0	0	0	0	1	
7	0	0	0	0	0	0	0	0

Poiché, come già detto, ogni riga della griglia 8*8 è in realtà un byte di memoria, ad ogni riga stessa corrisponderà un valore numerico dato dalla posizione degli 'uno' che vi si trovano. Per meglio chiarire questo concetto vediamo di fare un esempio.

Consideriamo la riga 0 della griglia di figura.

BIT → potenza di 2	7	6	5	4	3	2	1	∅
RIGA ∅	0	0	0	1	1	0	0	

Avremo:

2↑7*0 + 2↑6*0 + 2↑5*0 + 2↑4*1 + 2↑3*1 + 2↑2*0 + 2↑1*0 + 2↑0*0 = 16 + 8 = 24

Basta cioè usare la seguente formuletta: 2 ↑ posizione * bit elevare cioè 2 alla potenza data dalla posizione che stiamo considerando e moltiplicare il tutto per il valore del bit in quella posizione. Questo lo si deve fare per gli 8 bit del byte, poi si sommano i risultati di ogni operazione e si trova il valore da immettere in quel byte. (Consiglio, a chi non ha mai sentito parlare di queste cose di andarsi a leggere qualche articolo di aritmetica binaria).

A questo punto è chiaro che per creare il nostro 'mostriciattolo' basterà disegnarlo su di un foglio di carta, in una griglia come quella più sopra, ed ad ogni riga far poi corrispondere il valore numerico da inserire nella Ram assegnata (oppure servirsi del programma EDCAR).

Facciamolo:

	7	6	5	4	3	2	1	0	
0	*							*	: 128+1=129
1		*					*		: 64+2=66
2			*	*	*	*			: 32+16+8+4=60
3		*		*	*		*		: 64+16+8+2=90
4		*	*	*	*	*	*		: 64+32+16+8+4+2=126
5			*	*	*	*			: 32+16+8+4=60
6		*					*		: 64+2=66
7	*		*			*		*	: 128+32+4+1=165

Adesso che abbiamo fatto tutti i 'conticini', sperando di non aver sbagliato, possiamo scrivere il seguente programmino di prova:

```
10 for sa = 7168 to 7168+7
20 read ca
30 poke sa, ca
40 next sa
50 data 129, 66, 60, 90, 126, 60, 66, 165
```

Diamo il fatidico RUN, ma non accade nulla!! Niente paura. Digitiamo in modo diretto 'Poke 36869, (Peek (36869) or 15)', se ora premiamo alcune volte il tasto ' ' vedremo sul video il nostro tanto atteso mostriciattolo spaziale. Non abbiamo fatto altro che mettere in pratica le nostre conoscenze sul Vic (inteso non come il cervellone, ma come il Video Interface Chip).

Per praticità abbiamo riprogrammato il primo carattere della Mappa Caratteri, la ' ', ma potevamo benissimo sceglierne uno qualsiasi. Per far questo è però opportuno trovare il giusto indirizzo corrispondente a tale carattere. Un metodo può essere il seguente:

Posizione carattere = inizio mappa caratteri + 8* codice schermo carattere.

Inoltre abbiamo sostituito un solo carattere, ma ovviamente potremmo riprogrammare tutto il set dei caratteri standard e crearci così una serie di caratteri personalizzati, scrivendo magari in Greco, Cinese o Russo.

Nota: occorre tener presente che i nuovi caratteri possono essere usati sia con dei comandi di 'PRINT' che con dei 'POKE' di schermo ma in questo ultimo caso bisogna anche determinare il colore in cui lo vogliamo, poiché in caso contrario non vedremo niente, in quanto avremo una resa di bianco su bianco.

I PROGRAMMI

● *ROM in RAM LM.* È una breve routine in linguaggio macchina che consente di trasferire quasi istantaneamente la mappa dei caratteri della ROM nella RAM all'indirizzo 5120-($1400). La routine stessa si accorge se stiamo usando un VIC 20 con espansione maggiore ai 3k (riga 210) ed in questo caso l'indirizzo di inizio nella RAM sarà spostato automaticamente a 7168-$1c00.

La routine lm si posiziona in una zona di pagina 2 libera a partire da 673-($02A1) (linea 160).

Può essere usata in un programma come subroutine.

● *CARATTERI 16*8.* È una routine in Basic che permette di creare caratteri standard, ma in doppia altezza. In pratica copia ogni byte di un carattere per due volte nella RAM dove poi si troverà la nuova mappa caratteri.

Come già detto nell'articolo, per poter utilizzare tali caratteri, oltre a spostare l'inizio della mappa caratteri occorre variare anche il bit 0 del registro #4 posto a 36867-$9003. Questa routine è stata utilizzata nel programma Vic Magic.

● *VIC MAGIC.* È un programma che dimostra le varie possibilità offerte dalla manipolazione del chip 6561.

Utilizza la routine CARATTERI 16*8 (linea 630,700). Può essere un'ottima presentazione per un gioco di vostra creazione (vedi linea 560).

● *ISTOGRAMMI.* È una dimostrazione pratica dell'utilità della grafica in vari programmi. Si tratta di un programma che disegna sul video degli istogrammi di diverso colore, fino ad un massimo di 10.

All'inizio vengono chiesti in INPUT quanti istogrammi si vogliono disegnare, dopodiché è richiesto, per ogni istogramma, il valore numerico (Max. 160) ed infine si passa al grafico vero e proprio.

Il programma è scritto per la versione base o +3k, ma per farlo girare con espansioni maggiori ai 3k basta togliere la linea 700, sostituendola con la linea 710 (ovviamente eliminando la REM iniziale).

Dalla linea 110 alla 570 si trovano le varie subroutine utilizzate dal programma principale, che inizia alla linea 700, e che possono essere sfruttate per altri programmi di grafica 'VIChinga'.

● *CARATTERI A 90°* È una breve routine che permette di 'slittare' i normali caratteri della ROM a 90°, ovviamente trasferendoli nella RAM. Essendo stata realizzata volutamente in Basic, per consentire di comprendere a tutti il suo funzionamento, risente della lentezza di tale linguaggio. Per questo motivo invece di inclinare tutti i caratteri, è stata scritta in modo tale da lavorare solamente sui caratteri che vengono inseriti nelle linee DA-

TA (vedremo magari la versione in Lm sul prossimo numero).

In pratica non viene fatto altro che leggere il primo bit di ogni byte del carattere in questione e, con questi, formare il primo byte del carattere traslato. Il tutto è, ovviamente, fatto per ognuno degli otto bit, così da avere alla fine il nuovo carattere.

Può essere utile in vari programmi (con le dovute modifiche potrebbe essere inserita nel programma Istogrammi).

● *EDCAR.* È un programma di grande utilità pratica. Si tratta di un Mini Editor di caratteri, vi consente cioè di rendere facile l'ideazione dei vostri caratteri personali, senza dover perdere tempo in conti e sprecare un sacco di carta.

Potrete disegnare i vostri caratteri nel 'quadratone' 8*8, che vedrete in alto a sinistra dello schermo, dove si trova un cursore, simulato da programma (linee 600-670), che potrete muovere con i tasti CRSR (Alto-Basso-Destra-Sinistra). Con il tasto F1 potrete accendere un punto del vostro carattere, che vedrete in alta risoluzione sotto al quadratone, mentre con F7 lo potrete spegnere (linee 700-740). Inoltre sulla destra del 'quadratone' si troveranno i valori da 'pokare' negli indirizzi di RAM interessati, che vengono continuamente aggiornati (linee 760-820).

Per uscire dal programma basta premere il tasto F8 (± F7 + shift) e vi verrà richiesta la conferma con il tasto RETURN. Se invece di RETURN premete un altro tasto qualsiasi tornerete al programma principale.

Il programma di memoria gira su ogni configurazione di memoria. Buon 'lavoro' a tutti.

NOTA

I programmi utilizzano tutti un piccolo trucco che consente di non rendere necessario il trasferimento dei caratteri alfanumerici dalla ROM alla RAM. Infatti se abbiamo posto il generatore di caratteri in RAM e scriviamo una stringa alfanumerica in reverse vedremo con sorpresa, ma non tanto se si pensa a come vengono visualizzati i caratteri, che su video verranno stampati i normali caratteri contenuti nella ROM. Questo fatto è utilissimo per poter mescolare la bassa e l'alta risoluzione sul video.

Giovanni Beani

```
100 REM *********************
110 REM *    ROM IN RAM LM  *
120 REM *TRASFERIMENTO DATI *
130 REM *     PER  VIC 20   *
150 REM *********************
160 SY=673:REM INDIRIZZO DI PARTENZA DOVE MEMORIZZARE ROUTINE LM
170 READA:IFA>255THEN190
```

```
180 POKESY+I,A:I=I+1:CK=CK+A:GOTO170
190 IFCK-A<>0THENPRINT"ERRORE NEI DATA !":END
200 REM SE ESPANSIONE MAGGIORE DI 3K ALLORA
205 REM PONE MAPPA CARATTERI A 7168-(ESA 1C00)
210 IFPEEK(644)-PEEK(642)>28THENPOKE SY+3,28
220 SYS(SY):REM TRASFERISCE ROM CARATTERI IN RAM
230 PRINT"TRASFERIMENTO AVVENUTO"
240 END: REM SOFTWARE DI GIOVANNI BEANI
673 DATA 169,000
675 DATA 162,020:REM CARICA REGISTRO X CON BYTE ALTO
676 REM DI INDIRIZZO NUOVA MAPPA CARATTERI (20=5120)
677 DATA 160,128,133
680 DATA 167,133,169,134,170,132,168
687 DATA 168,177,167,145,169,200,208
694 DATA 249,230,168,230,170,166,170
701 DATA 224,044,208,239,096,5273
READY.
```

```
63500 REM ********************
63510 REM *  CARATTERI  16*8  *
63520 REM *                   *
63530 REM *  PER VIC 20  CON  *
63540 REM *                   *
63550 REM * ESPANSIONE O SENZA*
63560 REM ********************
63570 :
63580 ROM=32768:REM INIZIO ROM CONTENENTE CARATTERI
63590 SA=7168   :REM INIZIO NUOVA MAPPA CARATTERI
63600 CA=27     :REM CARATTERI DA TRASFERIRE
63610 :
63620 REM ** TRASFERIMENTO CARATTERI **
63630 FORI=SA TO(SA+16*CA)STEP2
63640 POKEI,PEEK(ROM)     :REM COPIA PER DUE VOLTE
63650 POKEI+1,PEEK(ROM)   :REM LO STESSO BYTE
63660 ROM=ROM+1           :REM PROSSIMO BYTE
63670 NEXT: REM GIOVANNI BEANI
63680 :
63690 REM ** ABBASSA PUNTATORE FINE MEMORIA **
63700 POKE55,0:POKE56,SA/256:CLR:RESTORE
63710 END
READY.
```

```
10 REM *****************************
20 REM *   ISTOGRAMMI PER VIC 20    *
30 REM *                            *
40 REM *   G. BEANI  &  L. BONCI    *
50 REM *                            *
60 REM *****************************
```

```
100 GOTO700
110 REM ** RIEMPIE CARATTERI IN RAM **
120 FORI=0TO20
130 FORJ=7TO0STEP-1
140 POKESA+I*8+J,255
150 NEXT J,I
160 :
170 REM ** RIEMPIE BYTE RIMANENTI **
180 FORI=0TO(NU/2)-1
190 I%=NU(I)/8:I%=8-(NU(I)-8*I%):REM CALCOLO BYTE IN I%
200 IFI%=8THEN240
210 FORJ=7TOI%STEP-1
220 POKESA+8*(I+21)+J,255
230 NEXT
240 NEXT
250 RETURN
260 :
270 REM ** PULISCE RAM CARATTERI **
280 FORI=SATO(SA+8*31)
290 POKEI,0
300 NEXT
310 RETURN
320 :
330 REM ** DISEGNA ASSI CARTESIANI **
340 POKEVD+484,43+128:POKECL+484,0:REM ORIGINE ASSI
350 FORI=21TO0STEP-1
```

```
360 POKEVD+I*22,33+128:POKECL+22*I,0:REM ASSE Y
370 POKEVD+485+21-I,45+128:POKECL+485+21-I,0:REM ASSE X
380 NEXT
390 RETURN
400 :
410 REM ** SCRIVE CARATTERI PER ISTOGRAMMI SU VIDEO **
420 CO=2:REM CODICE COLORE
430 H=0:REM CONTATORE VARIABILE INDICIZZATA NU
440 FORJ=2TONUSTEP2
450 YY=21:REM CONTATORE COLONNE VIDEO
460 IFNU(H)<8THEN520
470 FORI=0TO(NU(H)-8)STEP8
480 POKEVD+YY*22+J,21-YY:REM SCRIVE CARATTERI DA @ A NU(I)
490 POKECL+YY*22+J,CO:REM E LI COLORA
500 YY=YY-1
510 NEXT
520 POKEVD+YY*22+J,20+(J/2):REM SCRIVE ULTIMO CARATTERE
525 REM DIVERSO PER TUTTI GLI ISTOGRAMMI
530 POKECL+YY*22+J,CO
540 H=H+1
550 CO=CO+1:IFCO=8THENCO=2
555 REM CAMBIA COLORE PER PROSSIMO ISTOGRAMMA
560 NEXT
570 RETURN
580 :
700 VD=7680:CL=38400:REM INIZIO MAPPA SCHERMO E COLORE
710 REM VD=4096:CL=37888 PER ESPANSIONE MAGGIORE AI +3K
720 SA=7168:REM INIZIO MAPPA CARATTERI IN RAM
730 VA=15:REM VALORE 4 BIT NIBBLE DESTRA
735 :      REM DEL REGISTRO CONTROLLO #6 DEL 6561
740 MP=PEEK(36869):REM MEMORIZZA NORMALE
745 :                 REM VALORE REGISTRO CONTROLLO #6
750 POKE55,0:POKE56,SA/256:REM ABBASSA FINE MEMORIA
755 :REM PER PROTEZIONE CARATTERI IN RAM
760 PRINTCHR$(147):GOSUB 280:REM PULISCE
765 REM                         AREA RAM CARATTERI
770 POKE36869,MP:REM REGISTRO CONTROLLO #6 6561 NORMALE
780 PRINTCHR$(147);"QUANTI NUMERI";:INPUT NU:IFNU>10THEN780
790 PRINT
800 FORI=0TONU-1
810 PRINT"NUM■ ";I+1;
820 INPUT NU(I)
830 NU(I)=ABS(NU(I))
840 IFNU(I)>160THENI=I-1:PRINTCHR$(145);
850 NEXT
860 NU=ABS(NU)*2
870 PRINTCHR$(147):REM (C) GIOVANNI BEANI OTTOBRE 1984
880 POKE36869,((PEEK(36869) AND MP)OR VA)
885 REM PUNTA ALLA NUOVA MAPPA CARATTERI
890 POKE36879,25
900 GOSUB340
910 GOSUB420
```

```
920 GOSUB120
930 IF PEEK(653)<> 1 THEN930:REM ATTENDE TASTO 'SHIFT'
940 POKE36879,27
950 GOTO760
READY.
```

```
100 REM ***********************
110 REM *     VIC   MAGIC  PER  *
120 REM *                       *
130 REM * VIC 20 CON QUALSIASI *
140 REM *                       *
145 REM *     CONFIGURAZIONE    *
150 REM ***********************
160 :
170 C1$="█@@@":C2$="@"
180 RV$=CHR$(18):CL$=CHR$(147)
190 GOSUB 640
200 MP=PEEK(36869)
210 POKE36869,PEEK(36869)OR15
220 PRINT CL$;SPC(44);
230 GOSUB 610
240 FORI=0TO3
250 PRINT SPC(2);C2$;SPC(6);C2$;SPC(6);C2$
260 NEXT
270 GOSUB 610:REM (C) 1984 GIOVANNI BEANI
280 PRINT:PRINT
290 PRINTTAB(6);RV$;"COMMODORE"
300 PRINTTAB(4);RV$;"COMPUTER█CLUB"
310 PRINT:PRINT:PRINT:PRINT:PRINT:PRINT:PRINT:PRINT
320 PRINTTAB(6);RV$;" PRESENTA  "
330 FORI=0TO100
340 FORJ=7TO0STEP-1
350 VA%=RND(1)*255
360 POKE7168+J,VA%
370 NEXTJ,I
380 FORI=46TO2STEP-2
390 POKE36867,I
400 DL=25:GOSUB590
410 NEXT
420 PRINTCL$
430 PRINT:PRINT
440 PRINTTAB(8);"V█I█C"
450 PRINT
460 PRINTTAB(6);"M█A█G█I█C"
470 PRINT:PRINT
480 PRINT"DI█GIOVANNI█BEANI"
490 FORI=2TO24STEP2
500 POKE36867,I OR 1
510 DL=70:GOSUB590
520 NEXT
```

```
530 GET WW$: IF WW$=""THEN 530: REM ATTENDE PRESS. TASTO
540 POKE198,0:POKE36869,MP:POKE368S7,174
550 PRINTCL$;
560 :
570 REM INSERIRE QUI CARICAMENTO PROGRAMMA
575 REM PRINCIPALE DA NASTRO O DA DISCO !
580 END
590 FORJ=0TODL:NEXT
600 RETURN
610 PRINT SPC(2);C1$;SPC(3);C1$;SPC(3);C1$
620 RETURN
630 REM ** TRASFERIMENTO CARATTERI **
640 ROM=32768:SA=7168:CA=27
650 FORI=SATO(SA+16*CA)STEP2
660 POKEI,PEEK(ROM):POKEI+1,PEEK(ROM)
670 ROM=ROM+1
680 NEXT
690 POKE55,0:POKE56,SA/256
700 RETURN
READY.
```

```
100 REM ********************
110 REM *       EDCAR      *
120 REM *DI                *
130 REM *  GIOVANNI BEANI   *
140 REM *                  *
150 REM *TEL.0584/20670    *
160 REM *OGNI CONFIGURAZIONE*
170 REM ********************
180 :
190 POKE56,28:POKE52,28:REM ABBASSA FINE MEMORIA E INIZIO STRINGHE
200 MC=7168:REM INIZIO MAPPA CARATTERI IN RAM
210 MS=PEEK(648)*256:REM MAPPA SCHERMO
220 CS=38400:REM MAPPA COLORE
230 IFMS=4096THENCS=37888
240 MS=MS+22*2+1
250 CS=CS+22*2+1
260 XX=0:YY=0
270 REM ** AZZERA CARATTERE **
280 FORJ=0TO7:POKEMC+J,0:NEXT
290 POKE36869,PEEK(36869)OR(MC/1024)+8:REM PUNTA ALLA NUOVA MAPPA
300 REM ** CREAZIONE SCHERMO **
310 PRINT"█76543210      VALORI"
320 PRINT"█╭";:GOSUB450:PRINT"╮"
330 FORI=0TO7
340 PRINT"█|        |";I:REM 8 SPAZI FRA LE STANGHETTE
350 NEXT
360 PRINT"█╰";:GOSUB450:PRINT"╯"
370 PRINT:PRINT:PRINTTAB(4)"@"
380 PRINT
```

```
390 PRINT"■          :":PRINT
400 PRINTTAB(5)"■F1 = DISEGNA"
410 PRINTTAB(5)"■F7 = CANCELLA"
420 PRINTTAB(5)"■F8 = FINE"
430 GOSUB760
440 GOTO500
450 FORI=1TO8
460 PRINT"—";
470 NEXT
480 RETURN
490 REM ** LETTURA TASTIERA **
500 GETR$:IFR$=""THENGOSUB610:GOTO500
510 IFR$="■"THENIFYY<154THENYY=YY+22
520 IFR$="□"THENIFYY>0THENYY=YY-22
530 IFR$="■"THENIFXX<7THENXX=XX+1
540 IFR$="■"THENIFXX>0THENXX=XX-1
550 IFR$="■"THENCA=170:CO=2:GOSUB630:GOSUB700
560 IFR$="■"THENCA=160:CO=1:GOSUB630:GOSUB700
570 IFR$="■"THEN850
580 GOTO500
590 :
600 REM ** SIMULA CURSORE **
610 CA=PEEK(MS+YY+XX)
620 CO=PEEK(CS+YY+XX)
630 POKEMS+YY+XX,230:POKECS+YY+XX,0
640 FORJ=1TO60:NEXT
650 POKEMS+YY+XX,CA:POKECS+YY+XX,CO
660 FORJ=1TO60:NEXT
670 RETURN
680 :
690 REM ** ACCENDE O SPEGNE UN PUNTO **
700 BI=7-XX
710 BY=YY/22
720 IFR$="■"THEN740
730 POKEMC+BY,PEEK(MC+BY)OR2↑BI
731 PRINT"■":FORI=1TO11:PRINT:NEXT:PRINT"@@■■■@@":GOTO760:
    REM ACCENDE IL PUNTO
740 POKEMC+BY,PEEK(MC+BY)AND(255-2↑BI):REM SPEGNE IL PUNTO
750 REM ** AGGIORNA VALORI SU VIDEO **
760 PRINTCHR$(19):PRINT
770 FORJ=0TO7
780 V$=STR$(PEEK(MC+J))
790 V$=LEFT$(V$+"   ",4)
800 PRINT"■";TAB(15);V$
810 NEXT
820 RETURN
830 :
840 REM ** USCITA **
850 PRINT:PRINT:PRINT:PRINT:PRINT:PRINT:PRINT:PRINT:PRINT:PRINT:PRINT
860 MG$="■ RETURN PER CONFERMA"
870 GETR$:IFR$<>""THEN930
880 PRINTMG$
```

```
890 FORJ=0TO300:NEXT
900 PRINT"▢▣";MG$;"▣▢"
910 FORJ=0TO300:NEXT
920 GOTO870
930 IFR$<>CHR$(13)THEN500
940 PRINTCHR$(147);
950 MC=240
960 IFPEEK(648)=16THENMC=192
970 POKE36869,MC
980 END
READY.
```

```
100 REM ************************
110 REM *     "CARATTERI A 90° "  *
140 REM *                      *
150 REM *     PER VIC 20 CON     *
160 REM *QUALSIASI CONFIGURAZIONE*
170 REM ************************
180 :
190 CN=7168:REM INIZIO NUOVA MAPPA CARATTERI
200 POKE56,CN/256:POKE52,CN/256:CLR:REM ABBASSA PUNTATORI
205 REM FINE MEMORIA E INIZIO STRINGHE
210 CN=7168:READC$:IFC$="*"THEN360:REM LETTURA CARATTERI NEI DATA
220 CH=32768+8*(ASC(C$)-64):REM CALCOLO INDIRIZZO
225 REM IN ROM CARATTERE LETTO
230 CN=CN+8:REM INCREMENTO MAPPA CARATTERI RAM
240 :
250 REM ROUTINE 'SLITTAMENTO' CARATTERE
260 FORJ=0TO7
270 PK=0
280 FORI=0TO7
290 IF((PEEK(CH+I)AND(2↑J))/(2↑J))THENPK=PKOR2↑(7-I):REM VEDI ARTICOLO
300 NEXT
310 POKECN+J,PK:REM SCRIVE BYTE TROVATO NELLA NUOVA MAPPA CARATTERI
320 NEXT
330 GOTO210:REM LETTURA NUOVO DATA
340 :
350 REM STAMPA SU VIDEO CARATTERI CREATI
360 MC=PEEK(36869)
370 POKE36869,MCOR15:REM PUNTA ALLA NUOVA MAPPA (7168-$1C00)
380 RESTORE:PRINT"▟"
390 READC$: IFC$="*"THEN410
400 PRINTC$:GOTO390
410 IF PEEK(653)<>1 THEN 410:REM ASPETTA 'SHIFT'
420 POKE36869,MC:END:REM RIPOSIZIONA REGISTRO ED ESCE
430 :
440 REM LINEE DATA IN CUI INSERIRE I CARATTERI DA 'SLITTARE'
450 DATA A,B,C,D,E
460 DATA *:REM CARATTERE DI FINE LETTURA
READY.
```

Vivere di Rendita

Quante volte, dopo aver compilato una schedina, ci siamo messi a sognare di far 13 per poter, con i soldi vinti, vivere tranquillamente per tutta la vita? Ma siamo poi così sicuri di poter vivere tranquillamente per tutta la vita? Mi spiego meglio: siamo sicuri che ci basteranno per condurre una vita agiata?

Lo scopo di questo programma è quello di fornire una risposta a queste domande (possibilmente positiva!). Il programma si basa sul presupposto che la vincita venga depositata in banca e che ogni mese si prelevi una quota (QUOTA MENSILE), che ci permetta di vivere agiatamente e in base ad alcuni dati che gli forniremo ulteriormente, ci indicherà per quanto tempo potremo continuare a prelevare denaro dalla banca.

Come dati input, oltre all'importo della VINCITA e alla QUOTA MENSILE, il computer chiederà il tasso d'inflazione presunto e il tasso d'interesse bancario. Queste due richieste (INFLAZIONE e INTERESSE) si sono rese necessarie per rendere più attendibile il risultato fornito dal computer perché a causa dell'inflazione il costo della vita continua ad aumentare; nello stesso momento però, sul nostro deposito in banca matureranno degli interessi. Quando dovremo prelevare la quota successiva, non dovremo più prelevare la quota mensile da noi stabilita, bensì quella quota aumentata dell'inflazione maturata fino a quel momento e in banca ci troveremo la vincita aumentata degli interessi maturati anch'essi fino a quel momento ecc.

Figura 1: grafico che raffigura l'andamento del nostro conto corrente.

Per capire meglio quanto detto, si può far riferimento al grafico.

'A' indica la vincita depositata in banca. Il segmento 'AB' indica 'A' più gli interessi maturati in un mese. La retta che parte da 'D' rappresenta la quota che dovremmo prelevare ogni mese. A causa dell'inflazione, però (e del continuo aumento del costo della vita) dovremo prelevare sempre di più. Il segmento 'BC', indica la quota prelevata che, come abbiamo detto, cresce sempre più (BC magg. B1C1 magg. B2C2... DD1 magg. D1D2 magg. D2D3). Il punto 'P' rappresenta il punto in cui avremo prelevato l'intero deposito (interessi compresi) e la distanza tra 'O' e 'P1', indica il tempo necessario perché questo si verifichi.

Il computer trasformerà questa distanza in anni e mesi, poi la visualizzarà. Dopo aver inserito i dati, il computer chiederà se vogliamo il 'PRELIEVO MENSILE' cioè se vogliamo vedere quanto dovremmo in teoria prelevare ogni mese; se la risposta sarà affermativa, subito appariranno i prelievi in sequenza, altrimenti apparirà un 'OK' che indica che i dati intermedi non verranno mostrati.

Il programma è strutturato nel modo seguente. Tralasciando le prime righe, nelle quali è contenuta l'intestazione, dalla riga 70 alla 100, dopo un CLR, troviamo il blocco INPUT: verranno chiesti, rispettivamente, l'importo della vincita, l'inflazione presunta espressa in percentuale, l'importo dei prelievi mensili e il tasso d'interesse bancario, sempre espresso in percentuale.

Alla riga 110 troviamo il test di controllo per controllare appunto se l'importo del prelievo è minore rispetto a quello della vincita.

In caso contrario il computer stampa un messaggio che indica che si cerca di prelevare più di quanto non si sia potuto versare grazie al deposito. Nel Blocco 140-160, troviamo la proposta di visualizzazione dei prelievi mensili. Dalla 170 alla 190 è presente il blocco di visualizzazione del prelievo mensile.

Dalla riga 200 alla 270, troviamo la sezione contatori dei giorni per i calcoli degli interessi e dell'inflazione, le relative chiamate di subroutines per i suddetti calcoli e infine, i tests per vedere quanto abbiamo prelevato e quanto ci rimane ancora da prelevare. Nel Blocco 280-380, troviamo l'apertura delle periferiche d'output e l'uscita dei risultati. Infine troviamo il 'cuore' del programma e cioè la soubroutine di calcolo dell'inflazione e del montante esistente in banca e nella quale si fa uso della formula montante = capitale* (1+i) elevato alla t. 'i' rappresenta la percentuale, sia d'interesse che di aumento e 't' il tempo espresso in anni.

Il programma è corto e facile da digitare. Quindi in bocca al lupo per il 13!

Luca Cirillo

```
10 REM ***************************
20 REM *  LUCA CIRILLO (MONZA)    *
30 REM *                          *
40 REM *  PROGRAMMA PER VIVERE... *
50 REM *       DI RENDITA !!!     *
60 REM ***************************
70 CLR:INPUT"DEPOSITO ";VT:AV=VT
80 INPUT"INFLAZIONE PRESUNTA (%) ";IN
90 INPUT"PRELIEVI MENSILI ";PM:AM=PM
100 INPUT"INTERESSI BANCARI (%) ";IB
110 IFPM<VTTHEN140
120 PRINT"NON PUOI PRELEVARE PIU' DI "
130 PRINT"QUANTO HAI VINTO !!! ":END
140 PRINT"VUOI IL PRELIEVO MENSILE (S/N) ?"
150 GETA$:IFA$="N"THENN=0:PRINT"OK":A$="":GOTO200
160 IFA$<>"S"THEN150
170 N=1
180 PRINT"IL "INT(II/30)+1" MESE, ";
190 PRINT"PRELEVERAI £ ";INT(AM)
200 II=II+30:C=VT:I=IB/100:T=30/360
210 GOSUB390:VT=MT
220 C=PM:I=IN/100:T=II/360:GOSUB390
230 AM=MT:VT=VT-AM
240 IFII>36000THENZ=1
250 IFVT>0ANDN=1THEN180
260 IFVT>0THEN200
270 AA=INT(II/360):II=II-AA*360:MM=INT(II/30)
275 PRINT"PREMI UN TASTO ":
276 POKE 198,0:WAIT 198,1
280 PRINT"OUTPUT SU SCHERMO O CARTA (S/C) ?"
290 GETA$:IFA$="S"THENOPEN3,3:GOTO320
300 IFA$<>"C"THEN290
310 OPEN3,4
320 PRINT#3,"CON UNA VINCITA DI £ "AV","
330 PRINT#3,"PRELEVANDO CIRCA £ "PM" MENSILMENTE "
331 PRINT#3,"CON L'INFLAZIONE AL "IN"%"
332 PRINT#3,"E INTERESSI BANCARI AL "IB"%"
340 PRINT#3,"SI POTRA' VIVERE TRANQUILLAMENTE "
350 IFZ<>1THEN370
360 PRINT#3,"...PER TUTTA LA VITA !!! ":GOTO380
370 PRINT#3,"PER "AA"/A "MM"/M"
380 PRINT#3:CLOSE3:END
390 REM** SUBR E DI CALCOLO MONTANTE **
400 MT=C*((1+I)↑T):RETURN
READY.
```

TURBO TAPE

Chiunque abbia a disposizione, come memoria di massa, il solo registratore, avrà sicuramente notato la lentezza con cui è costretto a memorizzare o leggere i propri listati.

Sul numero 12 di C.C.C. abbiamo suggerito un sistema che consente di dimezzare i tempi di lettura di un programma ricorrendo all'alterazione di alcuni "puntatori".

Tale procedimento è però limitato alla sola lettura di un listato già registrato e, benché valido, non risolve del tutto il problema della velocizzazione del trasferimento di informazioni. È però possibile "bypassare" le routine originali Commodore (che intervengono in fase di lettura e registrazione) e servirsi di altre routine, nettamente più veloci, in grado di svolgere le stesse funzioni con una sintassi diversa e, quel che più conta, in un tempo inferiore.

Il programma in oggetto è il famoso TURBO TAPE di cui abbiamo ottenuto il diritto di diffusione e ne abbiamo approfittato per inserirlo subito nella rivista su cassetta COMMODORE CLUB N. 3.

È Nella cassetta di cui si parla non solo figura, primo, il TURBO TAPE ma tutti i programmi presenti sono stati "velocizzati" ricorrendo allo stesso programma. Ciò ha consentito di lasciare inalterato almeno per ora il prezzo della stessa confezione (L. 4800) che, in caso contrario, saremmo stati costretti a ritoccare a causa di un aumento del costo del nastro magnetico.

È infatti intuitivo che aumentando la velocità di registrazione diminuisce proporzionalmente la quantità di nastro necessaria per immagazzinare i dati.

Come funziona il TURBO TAPE.

Una volta caricato il programma è sufficiente digitare RUN ed il tasto return affinchè le routine vengano automaticamente allocate nella parte alta della memoria (intorno a 50.000).

Da quel momento la sintassi necessaria sarà identica a quella nota con l'unica differenza che il comando deve esser preceduto dal carattere di freccia a sinistra (←).

È ovvio che caricando programmi Basic, utilizzandoli e modificandoli, i comandi saranno sempre disponibili a meno che non si carichi un listato in L.M. (come spesso accade con i videogame), che occupi le stesse locazioni di memoria RAM.

Quale è la velocità con cui opera il Commodore 64 dotato del TURBO TAPE? Scopritelo da soli: neanche io, all'inizio, volevo credere che si potesse registrare e leggere alla stessa velocità del disco ...

Alessandro de Simone

MA TRA QUANTO TEMPO POTRO' COMPERARLO?

Leggendo i giornali e guardandosi un pò in giro, ci si può rendere conto di come i prezzi aumentino con un ritmo impressionante.

Se noi, ad esempio, cominciamo oggi a mettere da parte i soldi per poter in seguito acquistare un ciclomotore, quando avremo raggiunto la cifra necessaria, avremo una grossa delusione. "Il prezzo è aumentato del tot %, ci dispiace" ci sentiremo dire dal venditore. E con la coda tra le gambe, torneremo a casa a fare i calcoli per stabilire quanto ci manca per raggiungere la sperata quota... aumentata dell'inflazione maturata fino a quel momento. Lo scopo di questo programma è quello di prevenire questi inconvenienti. Esso si basa sul presupposto che i risparmi si versino in banca o meglio, si versi una quota costante mensilmente e che la banca applichi una certa percentuale di interesse.

Dopo aver dato il 'RUN', il computer chiederà il costo del bene, cioè il prezzo attuale di ciò che interessa acquistare, seguito dal tasso d'inflazione presunto, dalla quota mensile che si versa e dal tasso d'interesse bancario.

Conclusa questa prima fase, sul video apparirà un messaggio di attesa e dopo un lasso di tempo (la cui lunghezza dipende dal rapporto bene-versamenti), apparirà un secondo messaggio, precisamente una richiesta d'indirizzamento dei dati output.

Essi possono essere visualizzati o stampati. Nel primo caso bisognerà premere 'S', altrimenti 'C' e tutti i dati immessi appariranno su carta, seguiti dal TEMPO necessario per raggiungere la somma sperata (che non sarà la somma immessa, bensì aumentata dell'inflazione maturata fino a quel momento).

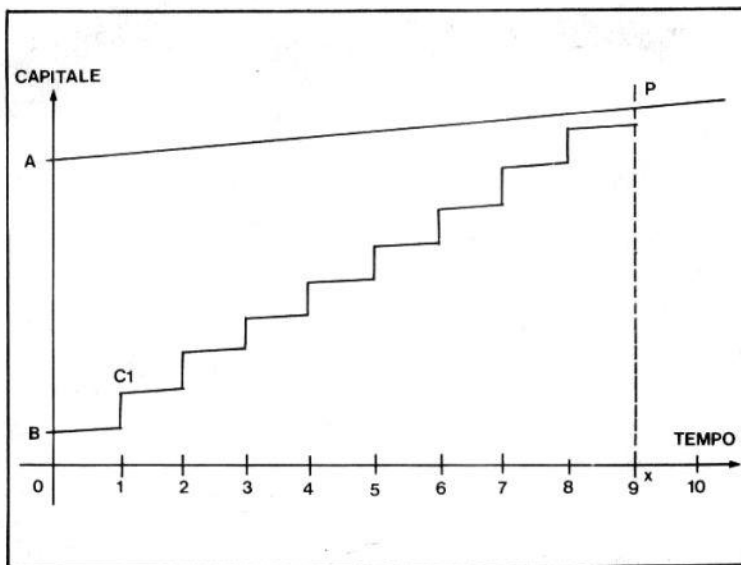



Figura 1: grafico che raffigura la nostra capacità di risparmio rispetto al costo di un bene. L'acquisto del bene è possibile a partire dal punto P.

Dal grafico pubblicato sarà possibile capire meglio quanto spiegato. Il costo del bene è rappresentato dalla retta avente origine in 'A' che, come possiamo notare, è in continuo aumento.

Il nostro primo versamento è rappresentato dal punto 'B'. Il segmento 'B'-'B1', rappresenta l'incremento che la quota (capitale) ha avuto col passare del tempo. Il segmento 'B1'-'C1' rappresenta il secondo versamento o meglio, il capitale dopo quest'ultimo versamento.

Se continuiamo ad incrementare il costo del bene e contemporaneamente i versamenti, si arriverà ad un punto (P) in cui il capitale in banca sarà uguale o addirittura maggiore rispetto al prezzo del bene: la differenza tra questo punto ('X' sul grafico) e l'origine ('O' sempre sul grafico), rappresenta il tempo necessario affinché l'evento si verifichi. Si potrebbe anche verificare il caso in cui il valore dell'inflazione sia maggiore rispetto alla somme del capitale esistente in banca maggiorato degli interesi; in questo caso la retta e la spezzata del grafico non si incontreranno mai: si parlerà allora di 'RETTE DIVERGENTI'. Per ovviare a questo inconveniente, bisognerà o aumentare i versamenti mensili o cercare una banca che applichi un tasso d'interesse maggiore o... scegliere un altro bene.

Comunque non è detto che si debba sempre verificare quest'ultimo evento (rette divergenti). Il computer calcolerà ogni

TELCOM DUE CARTE IN PIU'

LA NUOVA STAMPANTE JUKI A MARGHERITA
LA NUOVA STAMPANTE MITSUI AD AGHI

JUKI 6100

finalmente una stampante A MARGHERITA accessibile a tutti per il suo BASSO COSTO.
La caratteristica fondamentale di una stampante a margherita è la stampa a carattere pieno che garantisce una qualità di scrittura indispensabile per la corrispondenza automatica e il trattamento delle parole. L'alto prezzo di queste stampanti ne ha sempre impedito l'utilizzo in sistemi a basso costo quali personals, desk-top computers e micro in genere.

JUKI 6100 è la prima stampante a margherita che garantisce:

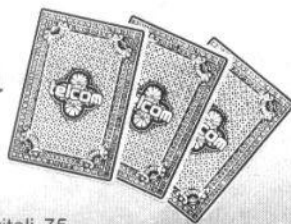
- prestazioni elevate
- affidabilità
- qualità di stampa
- prezzo coerente con i piccoli sistemi

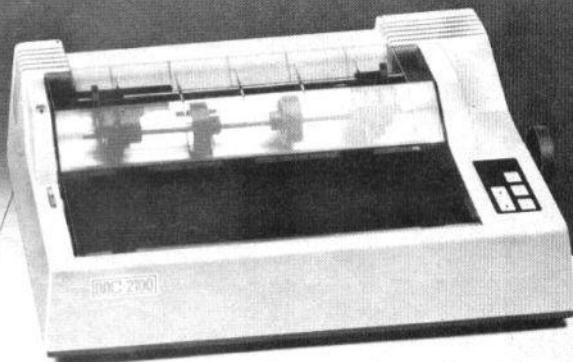
JUKI 6100: un passo avanti in tecnologia un passo indietro nel costo

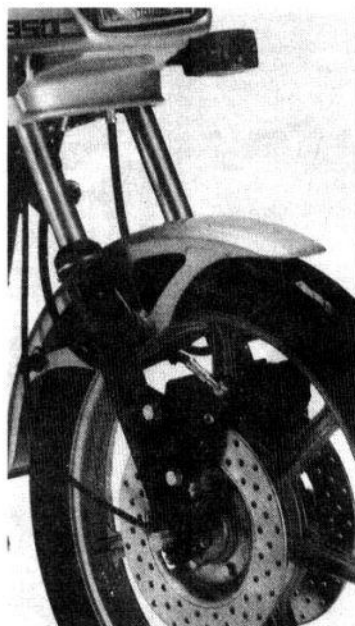
MITSUI Serie MC

Le stampanti della MITSUI rappresentano la continuità Telcom nella politica del "LOW-COST" con caratteristiche e qualità competitive:

- Velocità di 120/160 caratteri al secondo
- Testina a 9 aghi; 80/132 colonne
- Trascinamento per fogli singoli, per rotoli o per modulo continuo
- Interfaccia parallela e seriale
- Scrittura normale; espansa e NLQ
- Set di 95 caratteri
- Affidabilità elevata
- Ridotto costo di esercizio

MITSUI Serie MC: le stampanti "giuste" per micro e personals

Telcom s.r.l. - 20148 Milano - Via M. Civitali, 75
Tel. 4047648 (3 linee ric. aut.) - Telex 335654 TELCOM I

adverteam

giorno il valore del bene rivalutato (= inflazione) e il capitale esistente in banca, ne farà la differenza e... continuerà così, fino a quando avrà raggiunto il punto d'incontro o verificato il caso della divergenza; se si verificherà il secondo caso il computer fermerà l'esecuzione del programma, altrimenti tramuterà i giorni in mesi o addirittura in anni, dopo di che visualizzarà una richesta di indirizzamento dei dati, cioè se vorremo l'output su schermo o su carta: se premeremo 'S' avremo la visualizzazione dei dati, altrimenti la loro stampa.

Passiamo ora alla descrizione del programma:
tralasciando le prime 5 righe d'intestazione, troviamo, dopo un 'CLR', il blocco input con le richieste del 'COSTO BENE' ,'INFLAZIONE PRESUNTA' (in percentuale), 'VERSAMENTI MENSILI' e infine gli 'INTERESSI BANCARI' (sempre in percentuale). Alle righe 100-110 troviamo

la visualizzazione del messaggio d'attesa per l'elaborazione; alla 120 il primo test per controllare che i versamenti non siano maggiori del prezzo del bene, con relativa visualizzazione di un messaggio nel caso in cui si verifichi il caso. Nelle righe 140-180 troviamo la preparazione dei dati da inviare alla subroutine, il salto a quest'ultima, e il salvataggio in alcune variabili dei dati.

Alla riga 190 è presente il test per vedere se si è raggiunta la somma: in caso positivo si passerà alle successive righe di indirizzamento dei dati in output, altrimenti si salterà al confronto per vedere se le rette sono divergenti; in caso affermativo, si aumenterà il capitale in banca del nuovo versamento e si tornerà a confrontare gli importi. In caso negativo il programma si bloccherà. Infine alle righe 400-410 troviamo la subroutine di calcolo del capitale e dell'inflazione.

Il programma è facile e corto quindi... auguri per i futuri acquisti!!!

Luca Cirillo

```
10 REM ****************************
20 REM *     LUCA CIRILLO (MONZA)    *
25 REM *                              *
30 REM * FRA QUANTO TEMPO POTRO'     *
40 REM * ACQUISTARE LA MOTORETTA?   *
50 REM ****************************
60 CLR:INPUT"COSTO BENE ";CB
70 INPUT"INFLAZIONE PRESUNTA (%) ";IN
80 INPUT"VERSAMENTI MENSILI ";VM:AM=VM
90 INPUT"INTERESSI BANCARI (%) ";IB:N=30
100 PRINT"ATTENDERE PREGO !"
110 PRINT"STO ELABORANDO !!"
115 PRINT"(TIENI PREMUTO UN TASTO)"
120 IFCB>VMTHEN140
130 PRINT"VAI A COMPERARLO !!!"
131 PRINT"IL DENARO LO HAI GIA'":END
140 II=II+N:L=L+N:C=CB:I=IN/100
150 T=II/360:GOSUB400:A=MT
160 C=AM:I=IB/100:T=L/360
170 GOSUB400:A1=MT:
180 A2=A-A1
181 PRINT"DIFFERENZA"INT(A2)
182 PRINT"PREZZO BENE"INT(A)
183 PRINT"CIFRA IN BANCA"INT(A1):PRINT:PRINT
184 IFPEEK(197)=64THEN184
190 IFA2>0THEN310
191 POKE198,0:WAIT198,1
```

```
200 PRINT"OUTPUT SU SCHERMO O CARTA ? (S/C)"
210 GETA$:IFA$="S"THENOPEN3,3:GOTO240
220 IFA$<>"C"THEN210
230 OPEN3,4
240 PRINT#3,"IL BENE AVENTE COSTO DI £ "CB
250 PRINT#3,"VERSANDO MENSILMENTE £ "VM
255 PRINT#3,"AL TASSO D'INFLAZIONE DEL "IN"%"
256 PRINT#3,"ALL'INTERESSE BANCARIO DEL "IB"%"
260 AA=INT(II/360):II=II-AA*360:MM=INT(II/30)
270 II=II-MM*30:PRINT#3,"POTRAI COMPERARLO"
280 PRINT#3,"TRA "AA"/AA "MM"/MM "II"/GG.":
290 PRINT#3:CLOSE3:END
300 II=II-30:N=1:GOTO140
310 IFII=0THEN360
320 IF II>730THEN360
330 IFA2<A3THEN360
340 PRINT"ATTENZIONE !!!"
350 PRINT"RETTE DIVERGENTI !!!":END
360 A3=A2:IFZ=1THENZ=0:GOTO140
370 IFII<>INT(II/30)*30THEN140
380 Z=1:L=0:AM=MT
390 AM=AM+VM:A1=AM:GOTO180
400 REM ** SUBR E CALCOLO CAPITALE **
410 MT=C*((1+I)↑T):RETURN
READY.
```

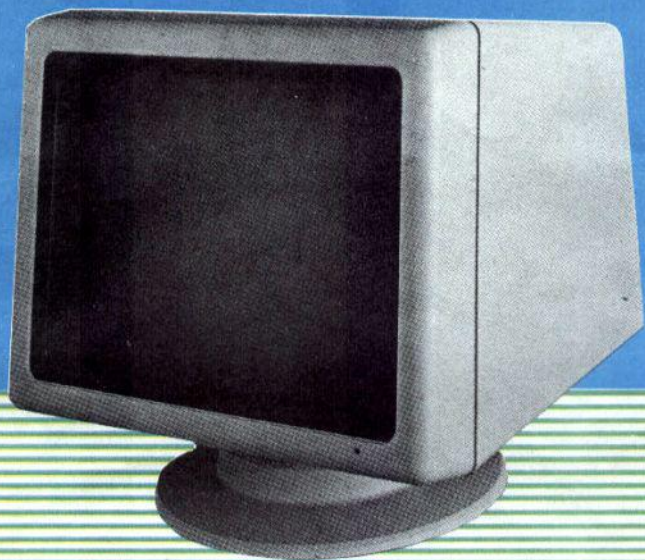
I MONITOR PRANDONI

Tutti coloro che comprano un personal computer lo collegano, in genere, ad un televisore domestico e si accorgono, dopo un po' di tempo, che questa forma di visualizzazione non è delle più felici. Abbiamo già avuto modo di dire che l'uscita video in alta frequenza (via UHF, tanto per intenderci) si effettua mediante un circuito che trasforma il calcolatore in una vera e propria stazione trasmittente, pur se di modestissima potenza, e, come tale, in grado di esser "ricevuta" da un qualsiasi televisore, proprio come se si trattasse di un telefilm di Dallas.

Con questo sistema, però, pur se si realizza una considerevole economia, la qualità dell'immagine è inevitabilmente modesta, appunto perché il segnale del computer è dapprima costretto ad esser "modulato" in modo da trasformarlo in un segnale ricevibile da un televisore, e poi nuovamente "demodulato" (all'interno del TV) per esser trattato e reso visibile. Con i due passaggi descritti, all'immagine si sovrappone un segnale, genericamente noto sotto il nome di "rumore", che è purtroppo inevitabilmente generato ogniqualvolta un segnale attraversa uno stadio elettronico.

Analogamente, un brano musicale ascoltato dal vivo appare decisamente diverso se riascoltato col registratore e, a maggior ragione, quasi incomprensibile se si registrá il brano con un secondo registratore che preleva il segnale dal precedente, in cascata.

Una immagine decisamente migliore si ottiene collegando il personal computer con la presa speciale che è ormai in dotazione dei moderni televisori. Attraverso di essa, infatti, si esclude la parte in alta frequenza e si elimina, tout court, l'introduzione del "rumore". A parte il fatto che il connettore descritto è di serie solo su alcuni modelli di punta dei moderni TV color, la qualità della visualizzazione non è delle migliori perché il TV nasce per soddisfare esigenze in cui la qualità elevata sarebbe uno spreco.

In un personal computer, però, la differenza si nota eccome, a seconda se si utilizza un comune TV, un TV dotato di ingresso speciale ed un monitor progettato appositamente per il collegamento con calcolatori. Numerose sono le differenze sulle quali potremmo soffermarci ma, per esser brevi e per non ricorrere a termini tecnici difficilmente comprensibili, esamineremo solo alcuni dei motivi

che pongono su di un piano nettamente diverso un monitor da un comune televisore.

Approfitteremo di queste note per presentare i monitor Prandoni.

● *La nitidezza.* I caratteri appaiono ben definiti, precisi nei contorni e perfettamente leggibili anche a distanza ravvicinata. L'articolo che state leggendo (e tantissimi altri) è stato scritto ricorrendo ad un word-processor e ad un monitor Prandoni che, anche dopo un numero prolungato di ore, non ha minimamente affaticato la vista.

● *I colori.* Questi sono ben definiti e il confine di due zone diversamente colorate è ben riconoscibile. Se un limite c'è, questo è dovuto unicamente ai limiti del segnale del computer adoperato (Commodore 64).

● *Uniformità della luminosità* e della densità del colore in ogni punto dello schermo. Le stesse videate, visualizzate con un comune TV color, presentavano sfocature ai bordi che, di scarsa importanza nel seguire programmi televisivi, costituiscono invece un difetto nel caso di utilizzo di W/P, Istogrammi eccetera.

I monitor Prandoni si presentano con una estetica davvero moderna e molto piacevole. La possibilità di orientare lo schermo nelle quattro direzioni, grazie ad un nodo sferico posto alla sua base, non può che essere apprezzato da coloro che sono costretti a porre l'unità video in zone troppo elevate o troppo basse a causa del limitato spazio disponibile sul tavolo da lavoro.

L'interruttore, posto al di sotto dello chassis, e i comandi di luminosità, colore e suono posti sul retro, partecipano a conferire quell'aspetto pulito, piacevole ed "essenziale" tanto di moda oggigiorno. Sul retro trovano posto la presa tripolare di alimentazione, la presa ad otto poli e due bipolari per gli ingressi. Non dimentichiamo, infatti, che ci troviamo di fronte ad un monitor di caratteristiche professionali che accetta in ingresso i segnali provenienti dagli standard più diffusi dei computer, tra cui il noto RGB in cui si prevedono tre segnali distinti per i tre colori fondamentali.

È ovviamente possibile la presa per il videoregistratore e, a richiesta, si forniscono i cavetti di collegamento per computer che non rispettano rigorosamente le convenzioni internazionali.

Alessandro de Simone

KING KONG

Scopo del gioco è condurre Mario dallo scimmione che si trova in cima alle scale. Questo per ben due volte, salendo per gli scalini e saltando i barili che rotolano giù: tutto va fatto in un solo minuto di gioco.

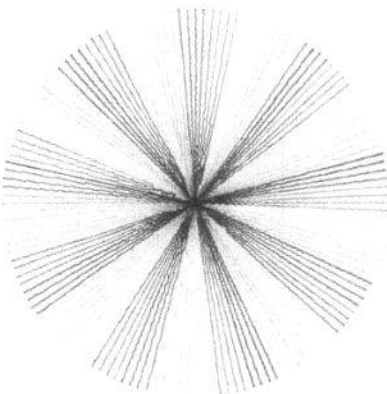
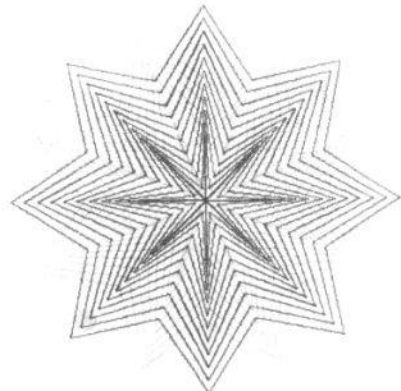
Gli ostacoli sono i barili che potete evitare saltandovi sopra guadagnando un punteggio per ogni salto. Un bonus viene assegnato ogni volta che raggiungete le cima.

Il gioco termina se scade il tempo stabilito, se un barile vi piomba addosso o se, saltando un barile, gli finite sopra senza scavalcarlo.

Andrea Campione

```
100 REM         KING KONG
110 REM PER VIC 20 INESPANSO
120 :
130 REM  DI ANDREA CAMPIONE
140 REM VIA DELLE RONDINI, 10
150 REM          CAGLIARI
160 REM   TEL. 070/371278
170 :
171 PRINT"USA I TASTI:"
172 PRINT"Z (SALTO)"
173 PRINT"< (SINISTRA)"
174 PRINT"> (DESTRA)"
175 PRINT"PREMI UN TASTO"
176 GETA$:IFA$=""THEN176
180 POKE36878,15
190 POKE36869,255:POKE36879,98
200 PRINT""
210 FORI=7168TO8185:POKEI,0:NEXT
220 FORI=7176TO7183:READA:POKEI,A:NEXT
230 FORI=7184TO7191:READB:POKEI,B:NEXT
240 FORI=7192TO7199:READC:POKEI,C:NEXT
250 FORI=7200TO7207:READD:POKEI,D:NEXT
260 FORI=7208TO7215:READE:POKEI,E:NEXT
270 FORI=7216TO7223:READF:POKEI,F:NEXT
280 READG
290 IFG=-1THEN320
300 POKEG,1
310 GOTO280
320 POKE7727,5:POKE7749,6
330 Q=8153:W=7750:R=7890:T=7968:Y=8086:P=000000:TI$="000000":V=2
340 POKEW,2:POKER,2:POKET,2:POKEY,2
350 POKE37154,127:NO=0
```

```
360 PRINT"1UP="P"      "TI$:GOSUB620
370 GOSUB620
380 IFW+1>8163THENPOKEW,0:W=7750
390 IFR+1>8163THENPOKER,0:R=7750
400 IFT+1>8163THENPOKET,0:T=7750
410 IFY+1>8163THENPOKEY,0:Y=7750
420 IFTI$="000100"THEN910
430 GOSUB620
440 IFPEEK(W+1)=4THEN910
450 IFPEEK(R+1)=4THEN910
460 IFPEEK(T+1)=4THEN910
470 IFPEEK(Y+1)=4THEN910
480 M=INT(10*RND(1))
490 IFM>5THENV=2:GOTO510
500 IFM<5THENV=3
510 IFPEEK(W+22)<>1THENPOKEW,0:POKEW+22,V:W=W+22
520 IFPEEK(R+22)<>1THENPOKER,0:POKER+22,V:R=R+22
530 IFPEEK(T+22)<>1THENPOKET,0:POKET+22,V:T=T+22
540 IFPEEK(Y+22)<>1THENPOKEY,0:POKEY+22,V:Y=Y+22
550 POKEW,0:POKER,0:POKET,0:POKEY,0
560 POKEW+1,V:POKER+1,V:POKET+1,V:POKEY+1,V
570 W=W+1:R=R+1:T=T+1:Y=Y+1
580 FORX=1TO10:NEXT
590 GOSUB620
600 IFNO>3THENPRINT"CONGRATULAZIONI!!":GOTO920
610 POKE8163,1:GOTO360
620 IFPEEK(197)=37THEN680
630 IFPEEK(197)=29THEN720
640 IFPEEK(197)=33THEN770
650 IFPEEK(Q+22)<>1THENPOKEQ,0:POKEQ+22,4:Q=Q+22
660 POKEQ,4
670 RETURN
680 IFPEEK(Q+1)=2ORPEEK(Q+1)=3THEN910
690 IFPEEK(Q+1)=1THENQ=Q-1
700 POKEQ,0:POKEQ+1,4:Q=Q+1
710 RETURN
720 IFPEEK(Q-1)=2ORPEEK(Q-1)=3THEN910
730 IFPEEK(Q-1)=1THENQ=Q+1
740 IFPEEK(Q-1)=6THENP=P+1000:POKEQ,0:NO=NO+1:Q=8153:RETURN
750 POKEQ,0:POKEQ-1,4:Q=Q-1
760 RETURN
770 IFPEEK(Q-22)=2ORPEEK(Q-22)=3THEN910
780 IFPEEK(Q-44)=1THENPOKEQ,0:POKEQ-22,4:Q=Q-22:GOSUB870:GOTO800
790 POKEQ,0:POKEQ-44,4:Q=Q-44:GOSUB870
800 IFPEEK(Q+44)=2ORPEEK(Q+44)=3THENP=P+100:PRINT"1UP="P"      "TI$
810 POKEQ,0:POKEQ-2,4:Q=Q-2:GOSUB870
820 IFPEEK(Q+45)=2ORPEEK(Q+45)=3THENP=P+100:PRINT"1UP="P"      "TI$
830 IFPEEK(Q+22)=1THENRETURN
840 IFPEEK(Q+22)=2ORPEEK(Q+22)=3THEN910
850 POKEQ,0:POKEQ+22,4:Q=Q+22:GOSUB870
860 GOTO820
870 POKE36874,128
```

```
880 POKE36874,0
890 FORZ=1TO10:NEXT
900 RETURN
910 PRINT"SQQQQQQQQQQQQQHAI PERSO!!!!!!!"
920 GETA$:IFA$=""THEN920
930 FORI=1TO1000:NEXT:RUN
940 DATA255,129,129,255,255,129,129,255
950 DATA60,126,189,219,231,219,189,60
960 DATA60,126,239,231,129,239,239,60
970 DATA24,26,10.30,104,8,20,34
980 DATA60,90,102,126,127,127,93,93
990 DATA93,93,28,127,99,99,227,227
1000 DATA7768,7769,7770,7771,7772,7773
1010 DATA7774,7775,7776,7777
1020 DATA7778,7779,7780,7781
1030 DATA7804,7805,7806,7807,7808
1040 DATA7831,7832,7833,7834,7835,7836
1050 DATA7859,7860,7861,7884,7885,7886,7887,7888
1060 DATA7911,7912,7913,7914,7937,7960
1070 DATA7983,7984,7985,7986,7987
1080 DATA7988,7989,7990,7991,8014,8015
1090 DATA8016,8017,8018,8019,8042,8043
1100 DATA8044,8045,8046
1110 DATA8069,8070,8071,8072,8073,8074,8075
1120 DATA8076,8077,8078,8079,8080,8081
1130 DATA8082,8083,8084
1140 DATA8107,8108,8109,8132,8133,8134
1150 DATA8135,8136,8137,8138,8139,8140,8141
1160 DATA8142,8143,8144,8145,8146,8147,8148
1170 DATA8171,8172,8173,8174,8175,8176,8177
1180 DATA8178,8179,8180,8181,8182,8183,8184,8185,-1

READY.
```

CONTATEMPO

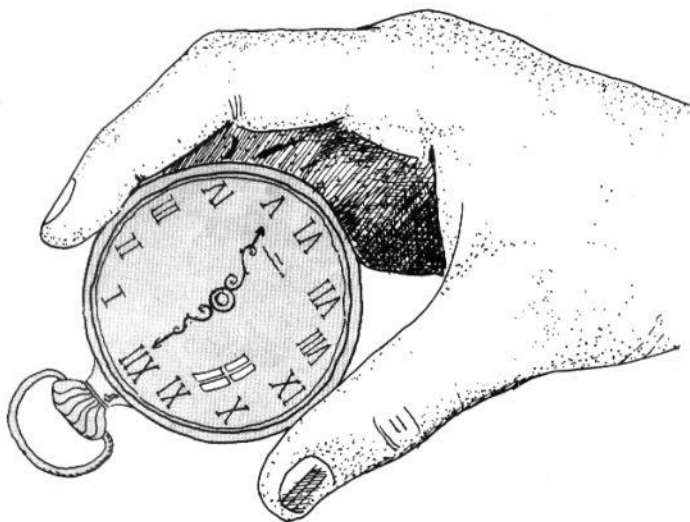
Un programma utile per avvisare oratori troppo chiacchieroni

In assemblee condominiali e scolastiche o in riunioni sindacali, fino ai più semplici consigli... familiari, capita spesso, purtroppo, che un oratore, convinto di dire cose molto interessanti, intrattenga il pubblico per troppo tempo creando disagi sia in coloro che devono parlare dopo di lui, sia nei suoi stessi sostenitori che si accorgono del tempo che passa ma non sanno come intervenire per farlo smettere.

Questo semplice programmino, fatto girare nella sala delle riunioni, consente di inserire in memoria i nominativi di cinque persone ed il tempo assegnato a ciascuna di esse.

Quando il programma "parte" un grosso orologio viene visualizzato sullo schermo segnando il tempo che passa. Non appena il tempo assegnato scade, un breve segnale sonoro viene emesso ad ogni secondo. L'oratore, dopo un po' di tempo, dovrebbe capire l'antifona anche perché il BIP, a lungo andare, diventa fastidioso.

Nessuno, quindi, si sentirà a disagio per avvisare il chiacchierone: scaricate la responsabilità dell'imbarazzo sul computer...

FAbio Sorgato

```
100 REM   CONTATEMPO PER COMMODORE 64
110 REM              BY
120 REM        FABIO  SORGATO
130 REM
160 REM
170 REM
180 A$(1)="..|...4..‐.|...|...|...|.‾‾‾‾"
190 A$(2)=",‾‾‾\|.. |.../../../../...‾‾‾‾"
200 A$(3)=",‾‾‾\... |.../..‾\... |\___/...."
210 A$(4)=".../../../../.. |‾‾‾|... |...."
220 A$(5)="|‾‾‾| ...‾‾‾\... |... |___/...."
230 A$(6)="../.../.../___.|..\|.. |\___/...."
240 A$(7)="|‾‾‾|... |.../.../.. |... |......"
250 A$(8)=",‾‾‾\|.. |\___//..\|.. |\___/...."
260 A$(9)=",‾‾‾\|.. |\___|... |... |\___/...."
270 A$(0)=",‾‾‾\|.. ||.. ||.. ||.. |\___/...."
```

```
280 REM TRASFORMA I NUMERI CONTEN. NEL VETT.A$ E LI PONE NEL VETT. B$
290 FORT=0TO9
300 FORK=1TO28:K$=MID$(A$(T),K,1)
310 IFK$<>"."THENC$(T)=C$(T)+K$:NEXT:NEXT:GOTO330
320 C$(T)=C$(T)+" ":NEXT:NEXT
330 FORT=0TO9:FORK=1TO28STEP4:B$(T)=B$(T)+MID$(C$(T),K,4)+"":NEXT
340 B$(T)=LEFT$(B$(T),LEN(B$(T))-5)+" ":NEXT
350 REM CHIEDE I NOMI E IL TEMPO.
360 INPUT"QUANTE PERSONE (MAX 5)";PR:IFPR<1ORPR>5THEN360
370 FORT=1TOPR:PRINT"NOME";T;CHR$(20);"A PERSONA":INPUTNO$(T):NEXT
380 PRINT"PER OGNUNO DI VOI DECIDETE IL TEMPO"
390 PRINT"A DISPOSIZIONE E DIGITATELO NEL FORMATO:"
400 PRINT TAB(17) "HHMMSS"
410 FORT=1TOPR
420 PRINT"IL TEMPO DI ";NO$(T);:INPUTT$(T)
430 IF LEN(T$(T))<>6THENPRINT"":GOTO420
440 FORK=1TO6:H$=MID$(T$(T),K,1)
450 IFVAL(H$)=0ANDH$<>"0"THENPRINT"":GOTO420
460 NEXT
470 FORK=1TO6STEP2
480 IFMID$(T$(T),K,2)>MID$("235959",K,2)THENPRINT"":GOTO420
490 NEXT:NEXT:POKE54296,10
500 REM ESEGUE IL CONTEGGIO
510 FORT=1TOPR:PRINT"ORA E' IL TURNO DI:":PRINT" "NO$(T)" "
520 PRINT"CHE DISPONE DEL TEMPO:"T$(T)""
530 PRINT"PREMI UN TASTO PER INIZIARE IL CONTEGGIO"
540 PRINT"ED UNO QUANDO PER INTERROMPERLO":POKE198,0:WAIT198,1:POKE198,0
550 TI$="000000":CK=0
560 PRINT""TAB(5);
570 FORJ=1TO6:PRINTB$(VAL(MID$(TI$,J,1)));:NEXT
580 IFTI$=>T$(T)THENGOSUB620
590 IFT$=TI$THEN590
600 T$=TI$:GETA$:IFA$=""THEN560
610 CT$(T)=TI$:NEXT:GOTO650
620 CK=1:POKE54276,0:POKE54277,0:POKE54278,0:POKE54277,9:POKE54278,0
630 POKE54276,17:POKE54273,40:POKE54272,38:RETURN
640 REM MOSTRA I TEMPI IMPIEGATI
650 PRINT"":FORT=1TOPR:PRINTNO$(T);" HA IMPIEGATO ";CT$(T)
660 PRINT"CONTRO IL TEMPO PREVISTO DI:"T$(T)
670 FORK=1TO40:PRINT"-";:NEXT:NEXT
680 PRINT"     PREMERE UN TASTO  PER CONTINUARE"
690 POKE198,0:WAIT198,1
700 MA=0:FORT=1TOPR
710 CV=VAL(CT$(T))-VAL(T$(T)):IFCV>=MATHENMA=CV:MA$=NO$(T)
720 NEXT:IFMA=0THEN760
730 PRINT"";MA$;" E' UNA PERSONA CHE CHIACCHIERA TROPPO!"
740 PRINT"HA SUPERATO PIU' DI TUTTI";
750 PRINT" IL TEMPO ASSEGNATOGLI."
760 PRINT"   PREMERE UN TASTO  PER RICOMINCIARE"
770 POKE198,0:WAIT198,1:POKE198,0:GOTO360

READY.
```

COMMODORE 64
UN REGALO
SPECIALE!

Tutto fantastico.
Scegli qui, fra tanti regali intelligenti, il più bel dono di Natale.
Per te, per un amico, per i tuoi figli.
Per giocare, studiare o lavorare.
Per inventare mille applicazioni creative.
Per accendere la fantasia, l'entusiasmo, il sorriso di chi sai tu. Sono proposte Commodore, il n. 1 dell'informatica.
COMMODORE:
Idea n. 1
Commodore 64.
Il più venduto nel mondo. Grande memoria, alta risoluzione grafica, sintetizzatore sonoro professionale, effetti tridimensionali. Scoprirai che mai un grande personal è costato così poco!
Idea n. 2
Il disk drive.
Potente, memorizza fino a 170.000 caratteri e accede in tempo reale agli archivi dati. Una sorpresa: è il meno caro che c'è.
Lire 630.000 + IVA
Idea n. 3
Registratore dedicato.
(Dedicalo a chi vuoi tu). Memorizza dati e programmi su normali cassette magnetiche.
Idea n. 4
Il plotter/stampante.
Una matita in mano al computer. Scrive, disegna, fa i grafici, stampa a 4 colori.
L. 375.000 + IVA
Idea n. 5
La stampante.
5 modelli tra cui scegliere.
Per stampare velocemente i tuoi programmi (anche a colori), corrispondenza, testi, libri.
Idea n. 6
Il monitor a colori professionale.
Schermo da 14'' ad alta risoluzione e audio incorporato.
L. 690.000 + IVA
Idea n. 7
Commodore 64 Executive.
Il personal portatile, un regalo per veri Executive. Monitor da 5'' a colori e disk drive da 170 K incorporati.
E un prezzo davvero speciale che solo Commodore può.
MPS 803
commodore
1520
64

TANTI REGALI SPECIALI!
COMMODORE 64
UN NATALE CHE VALE.

RIDEFINIZIONE DEI CARATTERI

Il programma facilita la programmazione di un nuovo set di caratteri. Per fare ciò all'inizio del programma viene trasferito il set di caratteri residente su ROM, tramite una routine in linguaggio macchina (vedi fig. 1), nella RAM a partire dall'indirizzo di memoria 12288.

La memorizzazione del nuovo set di caratteri avviene tramite l'utilizzo di una routine in linguaggio macchina (vedi fig. 2), mentre il caricamento dei dati relativi al nuovo set avviene tramite la routine rappresentata in figura 3.

```
*********************
*                   *
* SPOSTAMENTO DATI  *
*                   *
*********************

LDA #$0     ;
STA  $FB    ;
LDA #$D0    ;
STA  $FC    ;IMMAGAZ.IND.SORGENTE
LDA #$0     ;
STA  $FD    ;
LAD #$30    ;
STA  $FE    ;IMMAGAZ.IND.DESTINAZIONE
LDX #8      ;NUMERO DI BLOCCHI DA TRASF.
LDY #0      ;DIMENS.BLOCCO (256 PAROLE)
LDA ($FB),Y ;LEGGE UN ELEMENTO
STA ($FD),Y ;TRASFERISCE L'ELEMENTO
DEY         ;AGGIORNA PUNTATORE PAROLA
BNE  $F9    ;CONTR.TRASF. 256 PAROLE
INC  $FC    ;INCR.PUNTATORE SORGENTE
INC  $FE    ;INCR.PUNTATORE DESTINAZIONE
DEX         ;AGGIORNA IL CONT.DEL BLOCCO
BNE  $F2    ;CONTR.TRASF.BLOCCHI
RTS         ;RITORNA
```

Figura 1: trasferimento del set di caratteri da Rom a Ram.

```
********
*      *
* SAVE *
*      *
********

LDA #0     ;LUNGHEZZA NOME FILE
JSR $FFBD  ;CHIAMA ROUTINE SETNAM
LDA #1     ;NUMERO FILE LOGICO
LDX #1     ;MANDA I DATI AL REG.
LDY #1     ;INDIRIZZO SECONDARIO
JSR $FFBA  ;CHIAMA ROUTINE SETLFS
JSR $FFC0  ;CHIAMA ROUTINE OPEN
LDA $0     ;
STA $FB    ;
LDA $30    ;
STA $FC    ;IMMAGAZ.IND.SORGENTE
LDX $0     ;
LDY $40    ;IMMAGAZ.IND.DESTINAZIONE
LDA #$FB   ;IND.PAG.0 INIZ,TRASF,DATI
JSR $FFD8  ;CHIAMA ROUTINE SAVE
LDA #1     ;NUMERO FILE LOGICO
JSR $FFC3  ;CHIAMA ROUTINE CLOSE
RTS        ;RITORNA
```

Figura 2: routine per la memorizzazione del nuovo set di caratteri.

```
********
*      *
* LOAD *
*      *
********

LDA #0     ;LUNGHEZZA NOME FILE
JSR $FFBD  ;CHIAMA ROUTINE SETNAM
LDA #1     ;NUMERO FILE LOGICO
LDX #1     ;MANDA I DATI AL REG.
LDY #0     ;INDIRIZZO SECONDARIO
JSR $FFBA  ;CHIAMA ROUTINE SETLFS
JSR $FFC0  ;CHIAMA ROUTINE OPEN
LDA #0     ;PUNT.PER CARICAMENTO
LDX #$0    ;
LDY #$30   ;IND.PARTEZA BLOCCO DAT.
JSR $FFD5  ;CHIAMA ROUTINE LOAD
LDA #1     ;NUMERO DEL FILE LOGICO
JSR $FFC3  ;CHIAMA ROUTINE CLOSE
RTS        ;RITORNA
```

Figura 3: routine per il caricamento del nuovo set di caratteri.

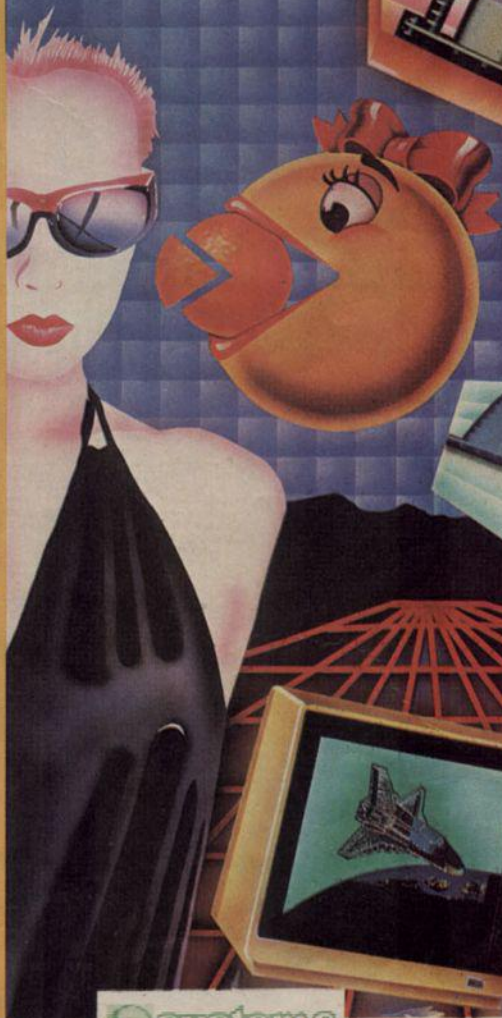
```
0 REM SET CARATTERI CBM64 LUCA DELLA VALLE
1 DATA169,0,133,251,169,208,133,252
2 DATA169,0,133,253,169,48,133,254
3 DATA162,8,160,0,177,251,145,253,136
4 DATA208,249,230,252,230,254,202,208
5 DATA242,96
6 DATA169,0,32,189,255,169,1,162,1
7 DATA160,1,32,186,255,32,192,255,165
8 DATA0,133,251,165,48,133,252,166,0
9 DATA164,64,169,251,32,216,255,169,1
10 DATA32,195,255,96
11 DATA169,0,32,189,255,169,1,162,1
12 DATA160,0,32,186,255,32,192,255,169
13 DATA0,162,0,160,48,32,213,255,169
14 DATA1,32,195,255,96
15 POKE53280,6:POKE53281,6
20 POKE650,128
25 POKE52,48:POKE56,48:CLR
30 FORI=49152TO49258
35 READA:POKEI,A:NEXT
36 GOTO10000
40 POKE56334,PEEK(56334)AND254
45 POKE1,PEEK(1)AND251
46 REM*******************
47 REM* SPOSTAMENTO DATI *
48 REM*******************
50 SYS(49152)
55 POKE1,PEEK(1)OR4
60 POKE56334,PEEK(56334)OR1
65 POKE53272,(PEEK(53272)AND240)OR12
70 X=2:Y=2
75 PRINT"
80 PRINT"┌─┬─┬─┬─┬─┬─┬─┬─┐"
85 FORI=0TO7
90 PRINT"│ │ │ │ │ │ │ │ │
95 PRINT"├─┼─┼─┼─┼─┼─┼─┼─┤
100 NEXT
105 PRINT"└─┴─┴─┴─┴─┴─┴─┴─┘
110 PRINT""TAB(28)"F1
115 PRINTTAB(20)"CARATTERE INFERIORE
120 PRINT""TAB(28)"F3"
125 PRINTTAB(20)"CARATTERE SUPERIORE
130 PRINT""TAB(28)"F5"
135 PRINTTAB(20)"ELABORAZIONE
140 PRINT""TAB(28)"F7"
145 PRINTTAB(20)"REGISTRAZIONE
150 PRINT"┌─┐
155 PRINT"│ │-CARATTERE IN ELABORAZIONE
160 PRINT"└─┘
165 PRINT"SPACE-PER FISSARE I PUNTI
170 PRINT"BAR SUL RETICOLATO
175 PRINT"←-PER CANCELLARE
180 AA=1024+X+40*Y:BB=32
185 POKE1786,B:POKE56058,4
190 POKE1024+X+40*Y,160:POKE55296+X+40*Y,14
195 FORI=0TO25:NEXT
200 GETA$:IFA$=""THENPOKE55296+X+40*Y,6:GOTO 185
205 IFA$=""THENB=B-1:IFB=-1THENB=255
210 IFA$=""THENB=B+1:IFB=256THENB=0
215 IFA$=" "THENBB=160:X=X+2
220 IFA$="←"THENBB=32:X=X+2
```

```
225 IFA$=CHR$(17)THENY=Y+2
230 IFA$=CHR$(29)THENX=X+2
235 IFA$=CHR$(145)THENY=Y-2
240 IFA$=CHR$(157)THENX=X-2
245 IFX>16THENX=2:Y=Y+2:IFY>16THENY=2
250 IFX<2THENX=16:Y=Y-2:IFY<2THENY=16
255 IFY<2THENY=2
260 IFY>16THENY=16
265 POKEAA,BB:AA=1024+X+40*Y:BB=PEEK(AA)
270 IFA$=""THEN285
275 IFA$=""THEN345
280 GOTO185
285 Z=0:Q=0
290 FORY=2TO16STEP2
295 FORX=2TO16STEP2
300 IFPEEK(1024+X+40*Y)=160THENQ=Q+2↑(8-(X/2))
305 NEXTX
310 A(Z)=Q:Z=Z+1:Q=0
315 NEXTY
320 E=12288+(B*8)
325 FORI=0TO7
330 POKEE+I,A(I)
335 NEXTI
340 GOTO70
345 PRINT"ARCHIVIAZIONE DEL NUOVO SET DI CARATTERI
346 REM********
347 REM* SAVE *
348 REM********
365 SYS(49187)
370 PRINT"OK !-IL NUOVO SET DI CARATTERI
375 PRINT"E' STATO MEMORIZZATO
380 PRINT"E N D":END
390 PRINT"INSERISCI LA CASSETTA CON I DATI
395 PRINT"DA CARICARE E PREMI UN TASTO QUALSIASI."
400 WAIT198,1
401 REM********
402 REM* LOAD *
403 REM********
405 SYS(49227)
410 POKE53272,(PEEK(53272)AND240)+12
415 PRINT"OK !!-IL NUOVO SET E'
420 PRINT"STATO INSERITO
430 NEW
10000 PRINT"M E N U '"
10005 PRINT"1] ELABORAZIONE NUOVO SET DI CARATTERI
10010 PRINT"E SUA MEMORIZZAZIONE SU CASSETTA."
10015 PRINT"2] CARICAMENTO SET DI CARATTERI DA CAS-
10020 PRINT"SETTA."
10025 PRINT"S C E G L I"
10030 GETA$:IF(A$<>"1"ANDA$<>"2")THEN10030
10035 A=VAL(A$):ON A GOTO 40,390
READY.
```

COMPUTER QUESTO MESE È QUESTO

COMPUTER

N. 72 - lire 3500 il "newsmagazine" dell'informatica

I packages grafici per il manager

Buongiorno, robot

Memorie di massa quali scegliere

Cristalli liquidi alla riscossa

Quinta generazione la sfida giapponese

Benchmark ICL modello 16

Antoni Francesco '84

systems

Computer - Anno VIII - n. 72 - 25 dicembre 1984 - 25 gennaio 1985 - Sped. Abb. Post. Gr. III/70 - Distr. MePe

COME OTTENERE I CARATTERI SPECIALI DEL COMPUTER COMMODORE

Codice	Tasti	Codice	Tasti
= [NERO]	CTRL + 1 BLK	= [ARANC]	C= + 1 BLK
= [BIANCO]	CTRL + 2 WHT	= [MARR]	C= + 2 WHT
= [ROSSO]	CTRL + 3 RED	= [ROSA]	C= + 3 RED
= [AZZUR]	CTRL + 4 CYN	= [GRIGIO1]	C= + 4 CYN
= [VIOLA]	CTRL + 5 PUR	= [GRIGIO2]	C= + 5 PUR
= [VERDE]	CTRL + 6 GRN	= [VERDE]	C= + 6 GRN
= [BLU]	CTRL + 7 BLU	= [CELESTE]	C= + 7 BLU
= [GIALLO]	CTRL + 8 YEL	= [GRIGIO3]	C= + 8 YEL
= [TF2]	f2	= [TF1]	f1
= [TF4]	f4	= [TF3]	f3
= [TF6]	f6	= [TF5]	f5
= [TF2]	f8	= [TF7]	f7
= [DOWN]	CRSR	= [UP]	SHIFT + CRSR
= [RVS]	CTRL + 9 RVS ON	= [RVOFF]	CTRL + 0 RVS OFF
= [HOME]	CLR HOME	= [CLEAR]	SHIFT + CLR HOME
= [RIGHT]	CRSR	= [LEFT]	SHIFT + CRSR
= [DEL]	INST DEL	= [PI]	SHIFT + ↑ π

IN MOLTI LISTATI PUBBLICATI IN QUESTA RIVISTA FIGURANO ALCUNI CARATTERI "SPECIALI" INDICHIAMO, SU QUESTA PAGINA, IL MODO DI OTTENERLI

STAR RACE

Un gioco scritto in Basic e linguaggio macchina che metterà a dura prova la vostra prontezza di riflessi.

Il programma gira sul Commodore 64 ed è composto da due parti: una in Basic e l'altra in LM.

La parte in Basic gestisce la lettura dei dati, il suono, i punteggi e i vari messaggi mentre la parte in LM è il gioco vero e proprio.

Lo scopo del gioco è quello di attraversare una zona dello spazio solcato da meteore che (guarda caso) cercano di venirci addosso.

Il movimento si ottiene mediante i tasti "<" (sinistra) e ">" (destra): bisogna cercare di resistere il più possibile in mezzo alla tempesta di meteore, che a mano a mano aumenta.

Si hanno tre possibilità, e se alla fine della terza si realizza il massimo punteggio il computer si congratula e tiene memorizzato tale punteggio. Il programma non ha la pretesa di essere particolarmente bello o divertente (infatti suono e grafica sono ridotti al minimo), ma fornisce un'idea di come possa realizzarsi un semplice gioco in LM.

```
K ?P

5 ::STAR RACE BY MARIANI G.
6 ::
10 :$1FF1 :LDA #$00    ;INIZIALIZZAZ.
11 :       :STA $03FE   ;
12 :       :LDA #$FA    ;
13 :       :STA $205E   ;
14 :       :LDA #$00    ;
15 :       :STA $03FF   ;
16 :       :LDA #$13    ;POSIZ. INIZIO
17 :       :STA $FA     ;
18 :$2004 :LDX $FA     ;CANCELLA GIOC.
19 :       :LDA #$20    ;
20 :       :STA $0608,X ;
21 :       :LDA $0630,X ;CONTROLLO OST.
22 :       :CMP #$20    ;
23 :       :BEQ $2013   ;
24 :       :RTS         ;RIT. A BASIC
25 :$2013 :LDA #$16    ;CARATTERE GIOC.
26 :       :STA $0630,X ;STAMPA GIOC.
27 :       :LDA #$01    ;COLORE GIOC.
28 :       :STA $DA30,X ;
29 :       :JSR $2047   ;
30 :       :LDA $C5     ;CONTROLLO TASTI
31 :       :CMP #$2C    ;TASTO '>'
32 :       :BEQ $202D   ;
33 :       :CMP #$2F    ;TASTO '<'
34 :       :BEQ $203A   ;
35 :       :JMP $2004   ;
36 :$202D :INC $FA     ;MUOVE A DESTRA
37 :       :LDA $FA     ;
38 :       :CMP #$1C    ;MARGINE DESTRO
39 :       :BNE $2004   ;
40 :       :DEC $FA     ;
41 :       :JMP $2004   ;
42 :       :DEC $FA     ;MUOVE A SIN.
43 :       :LDA $FA     ;
44 :       :CMP #$0B    ;MARGINE SIN
45 :       :BNE $2004   ;
46 :       :INC $FA     ;
47 :       :JMP $2004   ;
48 :$2047 :LDA #$5D    ;CARATT. STRADA
49 :       :STA $07CB   ;STAMPA BORDI..
50 :       :STA $07DC   ;..STRADA
51 :       :LDA #$01    ;COLORE STRADA
```

Insieme col listato basic, viene pubblicato anche un disassemblato commentato, per facilitare il lavoro a coloro che volessero modificarlo o migliorarlo, e permettere ai neofiti del LM di cominciare a capire le principali istruzioni sulle quali opera il linguaggio.

Il programma non è veloce come l'uso del LM farebbe supporre, perchè usa la routine di scroll del sistema operativo del 64 (vedi nel disassemblato JSR $E975), notevolmente lenta rispetto ad una realizzata dal programmatore.

Da notare che la locazione 2 contiene il numero di astronavi che rimangono da giocare, mentre $03FE-03FF contengono la difficoltà (progressiva) con la quale sta operando il programma.

Questo occupa i bytes da 8177 a 8340, mentre i caratteri sono a partire da 12288. Si raccomanda la solita attenzione nel trascrivere i dati perchè l'errore di un solo numero potrebbe bloccare il computer costringendovi a rifare tutto da capo. Per questa ragione si consiglia di registrare il programma prima di dare RUN in modo da averlo sempre a disposizione per eventuali correzioni o modifiche.

```
52 :         :STA $DBCB    ;
53 :         :STA $DBDC    ;
54 :         :DEC $03FF    ;ROUTINE PER..
55 :         :LDA $03FF    ;..INCREMENTO..
56 :         :CMP #$FA     ;..DIFFICOLTA'
57 :         :BNE $2076    ;
58 :         :LDA #$00     ;
59 :         :STA $03FF    ;
60 :         :LDA $D012    ;NUMERO RND..
61 :         :AND #$0F     ;..TRA 0 E 15
62 :         :TAX          ;
63 :         :LDA #$A0     ;CARATT.OSTACOLI
64 :         :STA $07CC,X  ;STAMPA OSTACOLI
65 :         :LDA #$07     ;COLORE OSTACOLI
66 :         :STA $DBCC,X  ;
67 :$2076   :JSR $E375    ;SCROLL NEL S.O.
68 :         :LDX $FA      ;
69 :         :LDA #$20     ;
70 :         :STA $0608,X  ;CANCELLA GIOC.
71 :         :DEC $03FE    ;CONTR. VARI..
72 :         :LDA $03FE    ;..PER LIVELLO..
73 :         :CMP #$00     ;..DIFFICOLTA'
74 :         :BNE $2094    ;
75 :         :LDA $205E    ;
76 :         :CMP #$FF     ;
77 :         :BEQ $2094    ;
78 :         :INC $205E    ;INC. DIFF.
79 :$2094   :RTS          ;RIT. DA SUB.

READY.
```

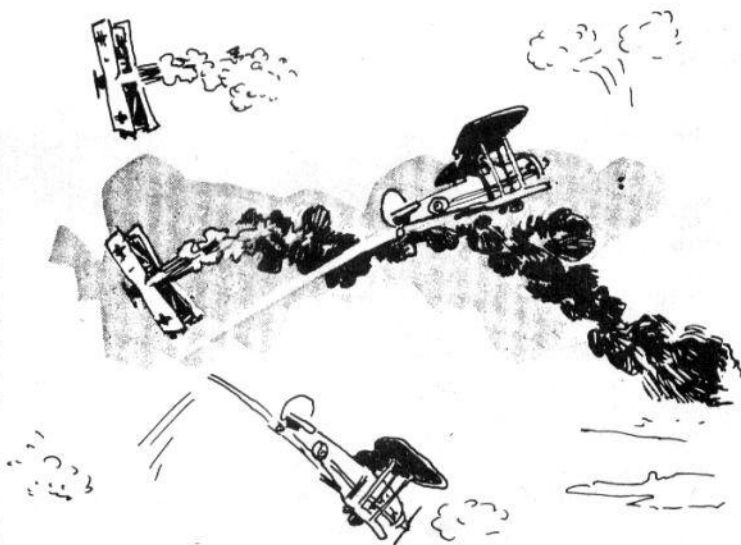
```
100 REM *** STAR RACE
110 REM *** BY MARIANI GIANCARLO
150 :
160 PRINT"[CLR]":RESTORE:POKE56,30:CLR:GOSUB410:B$="000000":P=53272
170 POKE54296,0:POKE53280,0:POKE53281,0:PRINT"[ ]":POKE2,3:P=53272
180 PRINT"[ ]                    STAR RACE"
190 PRINT"[ ]             BY MARIANI GIANCARLO"
200 PRINT"[ ]                TEL.  0362/72565"
210 PRINT"[ ]           TASTI DA USARE:   '<' E '>'"
220 PRINT"[ ]               HI SCORE:     "B$
230 PRINT"[ ]>>> PREMI UN TASTO  PER INIZIARE <<<"
240 POKE198,0:WAIT198,1:PRINT"[CLR]"
250 TI$="000000":POKEP,28:GOSUB490:SYS8177:POKE198,0
260 A$=TI$:GOSUB470:FORK=1TO1200:NEXT:POKEP,21
270 POKE2,PEEK(2)-1
280 IFPEEK(2)=0THEN350
290 PRINT"[ ]                 TI SEI SCHIANTATO !!"
300 PRINT"[ ]           ASTRONAVI  RIMANENTI: "PEEK(2)
310 PRINT"[ ]             >>> PREMI UN TASTO <<<"
```

```
320 POKE198,0:WAIT198,1:PRINT"":POKEP,28
330 TI$=A$:GOSUB490:SYS8187:POKE198,0
340 GOTO260
350 PRINT"                    FINE GIOCO"
360 PRINT"                 TEMPO:   "A$:IFA$>B$THEN390
370 PRINT"              >>> PREMI UN TASTO <<<":POKE198,0:WAIT198,1
380 PRINT"":GOTO170
390 PRINT"                CONGRATULAZIONI !!"
400 PRINT"             HAI FATTO L'HI-SCORE":B$=A$:GOTO370
410 U=0:FORK=8177TO8340:READA:POKEK,A
420 U=U+A:NEXT:IFU<>20948THENPRINT"ERRORE NEI DATI LM !":END
430 FORK=0TO23:READA:POKE12288+K,A:NEXT
440 FORK=12544TO12552:POKEK,0:NEXT
450 RETURN
460 :
470 POKE54276,0:POKE54277,0:POKE54278,0:POKE54277,74:POKE54278,3
480 POKE54276,129:POKE54273,10:POKE54272,37:RETURN
490 POKE54296,15
500 POKE54276,0:POKE54277,0:POKE54278,0
510 POKE54277,255:POKE54273,255:POKE54276,33
520 POKE54273,5:POKE54272,37:RETURN
530 :
540 REM *** PROGRAMMA LM ***
550 :
560 DATA 169,0,141,254,3,169,250,141
570 DATA 94,32,169,0,141,255,3,169
580 DATA 13,133,250,166,250,169,32,157
590 DATA 8,6,189,48,6,201,32,240
600 DATA 1,96,169,2,157,48,6,169
610 DATA 5,157,48,218,32,71,32,165
620 DATA 197,201,44,240,7,201,47,240
630 DATA 16,76,4,32,230,250,165,250
640 DATA 201,28,208,207,198,250,76,4
650 DATA 32,198,250,165,250,201,11,208
660 DATA 194,230,250,76,4,32,169,0
670 DATA 141,203,7,141,220,7,169,1
680 DATA 141,203,219,141,220,219,206,255
690 DATA 3,173,255,3,201,250,208,21
700 DATA 169,0,141,255,3,173,18,208
710 DATA 41,15,170,169,1,157,204,7
720 DATA 169,7,157,204,219,32,117,233
730 DATA 166,250,169,32,157,8,6,206
740 DATA 254,3,173,254,3,201,0,208
750 DATA 10,173,94,32,201,255,240,3,238,94,32,96
760 :
770 REM *** DATI PER CARATTERI ***
780 :
790 DATA 24,60,36,102,102,36,60,24
800 DATA 0,24,24,60,60,126,126,24
810 DATA 60,126,126,60,102,102,60,24

READY.
```

Guerre Atmosferiche

Un vecchio programma per il (glorioso) PET 2001 è stato "riciclato" per il Commodore 64. Le istruzioni sono descritte nelle righe da 2110 a 2290 che, se il lettore lo desidera, può omettere di digitare. L'unico accorgimento da seguire è quello di trascrivere con grande attenzione le righe dei DATA (270-280) che rappresentano una breve routine in linguaggio macchina. I più esperti, disassemblandola, possono inserirla in propri listati.

A che serve la routine? Scopritelo da soli...

A. De Simone.
Listato adattato da F. Sorgato

```
100 REM ********************************
110 REM **    GUERRE ATMOSFERICHE !    **
120 REM **                             **
130 REM **     PER  COMMODORE 64       **
140 REM ********************************
150 REM **  (EX - VERSIONE PET) BY     **
160 REM **        FABIO SORGATO        **
170 REM **                             **
180 REM **                             **
190 REM ********************************
200 :
210 PG$="             ":NM$="21":GOSUB2030
220 READAA,BB:FORZ=AATOBB:READW:POKEZ,W:NEXT
230 PRINT"██NOME DEL PRIMO GIOCATORE? ";:GOSUB1910:EP$=IN$
240 IFEP$=""THEN230
250 PRINT"█NOME DEL SECONDO GIOCATORE? ";:GOSUB1910:WP$=IN$
260 IFWP$=""THEN250
270 DATA826,849,162,4,160,0,132,33,134,34,177,33,73,128,145,33
```

```
280 DATA200,208,247,232,224,8,208,240,96,0
290 Q=59467:R=2:S=59464:POKEQ,16:POKE54296,10
300 POKES,0:H=1786:S9=54276:POKES9,32
310 REM DISEGNO FULMINE
320 A$=""
330 BL$="                                      "
340 C$=" / / \ \ ^_   \  / / "
350 D$="                 "
360 E$=C$+"\ / / /  / _/ / "
370 F$=D$+"             "
380 G$=C$+"\   \ \ \_  \"
390 H$=D$+"           "
400 I$=C$+"\ \  |  /  /  | "
410 J$=D$+"          "
420 PRINT"";
430 PRINT"r";:FORV=1TO38:PRINT"-";:NEXT:PRINT"┐";
440 F=28:G=28:M=3:MM=0
450 FORV=1TO19:PRINT"|";BL$;"|";:NEXT
460 REM DISEGNO CASETTE
470 PRINT"\ "SPC(20)" /"
480 PRINT" ┌────"SPC(20)"────┐ "
490 PRINT" |      "SPC(20)"      | "
500 T$=LEFT$(EP$,7):PRINT"";SPC(3.5+(7-LEN(T$))/2);T$
510 T$=LEFT$(WP$,7):PRINT"";SPC(30+(7-LEN(T$))/2);T$
520 PRINT" |_____"SPC(20)"_____|_"
530 PRINTA$:FORV=1TO40:PRINT"";:NEXT:PRINT
540 REM
550 GOSUB1350:IFF<1ORG<1THEN1710
560 M=M+1:IFM>3THENM=1:MM=MM+1:B=0:PRINT"";TAB(15);"────────────"
570 IFM=3ANDRND(1)>.33THEN560
580 PRINT"";:FORV=1TO3:PRINT"|";BL$;"|";:NEXTV:PRINT
590 A=1068:AA=RND(1)*26:A=A+AA:B=1-B
600 POKEA,104:FORV=1TO4:A=A+1:POKEA,102:NEXT
610 FORV=1TO5:A=A+1:POKEA,104:NEXT
620 A=A+27:POKEA,104:FORV=1TO11:A=A+1:POKEA,102:NEXT:POKEA,104:A=A+33
630 POKEA,104:FORV=1TO6:A=A+1:POKEA,102:NEXT:POKEA,104:NN=0
640 PRINTA$"";BL$;"  ":PRINTA$;"";
650 IFM=3THENPRINT"          "SPC(21)"          ":GOTO680
660 IFB=0THENPRINT"          "SPC(21)"ATTACCANTE":GOTO680
670 PRINT"ATTACCANTE"SPC(20)"          "
680 EE=INT(RND(1)*100)
690 IFEE>50THENGOSUB770:GOTO710
700 PRINTA$SPC(14)"VENTO ->"EE"      "
710 PRINTBL$:PRINTBL$;""
720 IFM<3THEN780
730 FORX=1TO4:PRINTBL$;"":GOSUB1630:GOSUB1610
740 PRINTTAB(13)"EVENTO CASUALE":GOSUB1630:NEXT
750 GOSUB1620:GOSUB1620:PRINTBL$
760 O=1:B$=MID$("PGTF",INT(RND(1)*4)+1,O):IN$=B$:GOTO860
770 EE=INT(-EE/2):PRINTA$SPC(14)"VENTO <-"ABS(EE)"      ":RETURN
780 IFM<>1THENGOSUB1630:GOTO830
790 POKES9,33:N9=220
```

```
800 PRINT"■";TAB(13)"BATTAGLIA"MM"■——":FORX=1TO2:GOSUB2310
810 GOSUB1610
820 NEXT:POKES9,32
830 T$=EP$:IFB=0THENT$=WP$
840 PRINTA$"■■";T$;": TEMPO  (G,F,P,T,U) ? ";:GOSUB1910:PRINT"□";BL$
850 B$=LEFT$(IN$,1):O=LEN(IN$):IFO=0THEN840
860 W$="PIOGGIA":IFLEFT$(W$,O)=IN$THENFF=5:TT=FF:O=2:GOTO920
870 W$="GRANDINE":IFLEFT$(W$,O)=IN$THENCC=58:FF=4:TT=FF:O=0:GOTO920
880 W$="TORNADO":IFLEFT$(W$,O)=IN$THENCC=102:FF=7:TT=FF:O=1:GOTO920
890 W$="FULMINE":IFLEFT$(W$,O)=IN$THENO=0:GOTO920
900 W$="USCITA":IFLEFT$(W$,O)=IN$THEN1710
910 PRINT"□□":GOTO710
920 PRINT"□COLPISCE CON ■";W$
930 IFO=1THENPRINT"■■";:FORV=1TO40:PRINT"■";:NEXT:PRINTA$;"■■
940 IFM=3THENA1=0:PRINT
950 IFM<3THENPRINT"■CORREZIONE? ";:GOSUB1910:BB$=IN$:A1=VAL(BB$)
960 PRINT"□LA CORREZIONE E'";A1
970 IFA1<-150THENA1=-150
980 IFA1>150THENA1=150
990 IFB$="G"THENGOSUB1620:GOTO1020
1000 IFB$="F"THENGOSUB1620:GOTO1410
1010 IFO=1THENPRINT"■■■■";:FORV=1TO20:PRINT"■";:NEXT:PRINT:EE=1.5*EE
1020 E=(A1+EE)/50:DD=0:GG=104
1030 IFO=2ANDE<0THENCC=78:GOTO1060
1040 IFO=2ANDE>0THENCC=77:GOTO1060
1050 IFO=2THENCC=66
1060 A=1148+AA:BB=0:FF=TT:T=WW:I=0:POKES9,33
1070 FORW=1TO16:WW=W:N9=W*10:GOSUB2310
1080 IFO<>1ORDD=1THENPOKES9,32
1090 IFT=WWTHENW=16:WW=0:REM A. DE SIMONE
1100 A=A+(40+E):GOSUB1240:NEXT:POKES9,32
1110 IFO<>1THEN1200
1120 IFPEEK(C+1)=93ORPEEK(C-1)=93THEN1200
1130 POKEC+1,GG:POKEC-1,GG
1140 IFPEEK(C+2)=93ORPEEK(C-2)=93THEN1200
1150 POKEC+2,GG:POKEC-2,GG
1160 IFE>0ORE=0THENPOKEC-38,32:POKEC-39,32
1170 IFE<0ORE=0THENPOKEC-41,32:POKEC-42,32
1180 IFDD=0THENGOSUB1690:GOTO1200
1190 IFDD=1THENPOKES9,33:FORN9=200TO100STEP-2:GOSUB2310:NEXT:POKES9,32
1200 IFNK<>1THENGOSUB1620
1210 IFDD=0THENDD=1:CC=32:GG=CC:GOTO1060
1220 GOTO550
1230 GOTO 550
1240 C=A:FORZ=1TOFF:C=C+1:D=PEEK(C):IFO=1ANDE<0THEND=PEEK(C-1)
1250 IFD=93THENGOSUB1660:W=16:O=0:RETURN
1260 IFD>127ANDDD=0THENGOSUB1310
1270 IFI>2THENW=16
1280 POKEC,CC:IFCC=58THENN9=180:POKES9,33:GOSUB2310:POKES9,0
1290 NEXT:BB=1-BB:IFBB=0ANDB$="T"THENFF=FF-1
1300 RETURN
1310 FORX=1TO3:POKEC,170:GOSUB1340:POKEC,58
```

```
1320 GOSUB1340:NEXT:I=I+1:IFO<>0THENW=16
1330 RETURN
1340 POKES9,33:N9=150:GOSUB2310:FORV=1TO50:NEXT:POKES9,32:RETURN
1350 F=0:G=-1:P=0
1360 FORY=1TO4:FORX=1TO7:H=H+1:D=PEEK(H):IFD<>32THENP=P+1
1370 NEXT:H=H-47:NEXT
1380 IFG=-1THENF=P:G=0:P=0:H=H+187:GOTO1360
1390 G=P:H=1786
1400 RETURN
1410 A1=A1/33:IFA1<-4THENA1=-4
1420 IFA1>4THENA1=4
1430 IFA1>1THENA1=A1-1
1440 AA=AA+A1+7
1450 IFAA>33THENAA=33
1460 IFAA<6THENAA=6
1470 FORZ=1TO3:GOSUB1600:PRINTC$:SYS826:SYS826:GOSUB1600:PRINTD$:NEXT
1480 GOSUB1610:GOSUB1600:PRINTC$:SYS826
1490 SYS826:GOSUB1600:PRINTD$:GOSUB1620
1500 IFA1>0THEN1550
1510 IFA1=0THEN1580
1520 FORZ=1TO2:GOSUB1600:PRINTE$:SYS826:GOSUB1640:SYS826:GOSUB1600:PRINTF$
1530 AA=AA+1
1540 NEXT:GOTO550
1550 FORZ=1TO2:GOSUB1600:PRINTG$:SYS826:GOSUB1640:SYS826:GOSUB1600:PRINTH$
1560 AA=AA-1
1570 NEXT:GOTO550
1580 FORZ=1TO2:GOSUB1600:PRINTI$:SYS826:GOSUB1640:SYS826:GOSUB1600:PRINTJ$
1590 NEXT:GOTO550
1600 PRINT"■■■■■"SPC(AA);:FORV=1TO25:NEXT:RETURN
1610 FORV=1TO500:NEXT:RETURN
1620 FORV=1TO1000:NEXT:RETURN
1630 POKES9,33:N9=200:GOSUB2310:FORV=1TO100:NEXT:POKES9,32:RETURN
1640 REM
1650 RETURN
1660 IFNN=1THENRETURN
1670 PRINT"■";TAB(13);"■ FUORI CAMPO!!":GOSUB1690:FORV=1TO2500:NEXT
1680 PRINT"■";TAB(13)"BATTAGLIA"MM"■——":NN=1:RETURN
1690 POKES9,33:FORX=1TO10:FORV=10TO200STEP25:N9=V:GOSUB2310:NEXT:NEXT
1700 POKES9,32:RETURN
1710 PRINTA$;"■  ";BL$:PRINTBL$:PRINTBL$
1720 FORX=1TO500:NEXT
1730 POKES9,33:FORV=200TO50STEP-1:N9=V:GOSUB2310:NEXT:POKES9,32
1740 B$=WP$:P=F*3.6:IFG>0THENB$=EP$:P=G*3.6
1750 T=7-(LEN(EP$)/2):IFT<0THENT=0
1760 PRINT"■■■■■";TAB(T);EP$
1770 T=33.5-LEN(WP$)/2:IFT+LEN(WP$)>39THENT=40-LEN(WP$)
1780 PRINT"■■■";TAB(T);WP$
1790 PRINT "■■■■■■■■■■";INT(F*3.6);"■% E' SALVO"
1800 PRINT "■■";TAB(26);INT(G*3.6);"■% E' SALVO"
1810 PRINT"■■■■■■■■■■■■■■■■■■■■■■■■■■";
1820 IF F=G THENPRINT"E' UN PAREGGIO!!":GOTO 1850
1830 PRINT"IL VINCITORE E'";:IF F<G THEN PRINT WP$:GOTO 1850
```

```
1840 PRINT EP$
1850 PRINTA$;"█▓GIOCATE ANCORA? ";:GOSUB1910
1860 IFLEFT$(IN$,1)="N"THEN1890
1870 T$=EP$:EP$=WP$:WP$=T$
1880 GOTO 420
1890 PRINT"▀█▓":END
1900 END
1910 IN$="  ":ZT=TI:ZC=2:ZD$=CHR$(20)
1920 GETZ$:IFZ$<>""THEN1950
1930 IFZT<=TITHENPRINTMID$(" ▒",ZC,1);"▌";:ZC=3-ZC:ZT=TI+15
1940 GOTO1920
1950 Z=ASC(Z$):ZL=LEN(IN$):IF(ZAND127)<32THENPRINT" ▌";:GOTO1990
1960 IFFLAND(ZAND127)>64AND(ZAND127)<91THENZ$=CHR$((Z+128)AND255)
1970 IFZL>254THEN1920
1980 IN$=IN$+Z$:PRINTZ$;ZD$;Z$;
1990 IFZ=13THENIN$=MID$(IN$,2):PRINTCR$;:RETURN
2000 IFZ=20ANDZL>1THENIN$=LEFT$(IN$,ZL-1):PRINT"▌";:GOTO1920
2010 IFZ=141THENZ$=CHR$(-20*(ZL>1)):FORZ=2TOZL:PRINTZ$;:NEXTZ:GOTO1910
2020 GOTO1920
2030 PRINT"▀":FORI=1TO10:PRINT"————";:NEXTI:PRINT
2040 PRINTTAB(6)"COLPISCI IL NEMICO"
2050 PRINT"CON LA FORZA DELLE INTEMPERIE!"
2060 PRINT:PRINT:PRINT:PRINT
2070 PRINT"VUOI LE ISTRUZIONI? (S/N)"
2080 GETA$:IFA$=""THEN2080
2090 IFA$="S"THENGOSUB2110
2100 PRINT"▀":CLR:CR$=CHR$(13):GOTO220
2110 PRINT"▀ QUESTO GIOCO SIMULA UNA BATTAGLIA TRA
2120 PRINT"DUE AVVERSARI, RAPPRESENTATI  DA  DUE
2130 PRINT"CASETTE, CHE USANO COME ARMI...EVENTI
2140 PRINT"ATMOSFERICI.":PRINT" ALL'INIZIO DEL GIOCO APPARE UNA NUVOLA
2150 PRINT"POSIZIONATA  IN  MODO CASUALE ED UN MES-▌
2160 PRINT"□SAGGIO CHE INDICA LA DIREZIONE DEL VENTO
2170 PRINT"□E LA SUA VELOCITA'. IL GIOCATORE INDICA-
2180 PRINT"□TO COME ▓ATTACCANTE▄ DEVE SCEGLIERE UNA
2190 PRINT"DELLE QUATTRO ARMI DELLA NATURA CHE SONO▌
2200 PRINT"□A SUA DISPOSIZIONE:
2210 PRINT"█▓G=GRANDINE▄ ▓P=PIOGGIA▄ ▓T=TORNADO▄ ▓F=FULMINE▄
2220 PRINT"U=USCITA DAL GIOCO.(FINE)
2230 PRINT:PRINT"EGLI DEVE INOLTRE INDICARE, SE LO DESI-
2240 PRINT"DERA, LA CORREZIONE DA APPORTARE ALLA
2250 PRINT"DIREZIONE DEL VENTO, TENENDO CONTO CHE
2260 PRINT"UN NUMERO NEGATIVO RAPPRESENTA UNO SPO-
2270 PRINT"STAMENTO A SINISTRA, ED UNO POSITIVO UNO
2280 PRINT"□SPOSTAMENTO A DESTRA.
2290 GETA$:IFA$=""THEN2290
2300 PRINT"▀":RETURN
2310 POKE54276,0:POKE54277,0:POKE54278,0
2320 POKE54277,0:POKE54278,128:POKE54276,33
2330 POKE54273,N9:POKE54272,0:RETURN

READY.
```

VITA

Un modello del mondo ed un mondo in se stesso

Lo spunto per il programma pubblicato, è stato fornito dalla lettura della rubrica *(Ri) creazioni al calcolatore* sulla rivista *Le Scienze* (ed. Italiana di *Scientific American* n. 189 maggio 1984).

Il gioco *Vita* è stato inventato nel 1970 da John Horton Conway. Esso ha appunto un aspetto biologico, come del resto suggerisce il nome. Le cellule nascono, vivono o muoiono a seconda della densità della popolazione locale. "L'Universo", su cui queste evolvono, è una griglia rettangolare, dove ogni sito può, o meno, essere occupato da una cellula.

A partire da una configurazione iniziale, si effettua, per passi successivi, un controllo su ogni cella e sulle otto celle che si formano intorno. (Si veda la figura 1), al fine di calcolare il nuovo stato della cella nella generazione successiva. Se la cellula centrale è viva, (cioè se il sito è occupato), essa continuerà a vivere a condizione che attorno a lei ci siano 2 oppure 3 cellule vive. Negli altri casi muore, o per solitudine o per sovraffollamento.

Se invece la cella centrale è vuota, rimarrà vuota a meno che il suo intorno contenga esattamente 3 cellule vive, nel qual caso si ha la generazione di una nuova cellula. Come si vede le regole di composizione sono molto semplici, anche se il gioco ha il fascino dell'imprevedibilità. Alcune configurazioni terminano con la morte di tutte le cellule, ma molto più spesso si finisce in configurazioni stabili, oppure periodiche.

L'implementazione di quanto detto, sul nostro C64, è stata fatta utilizzando

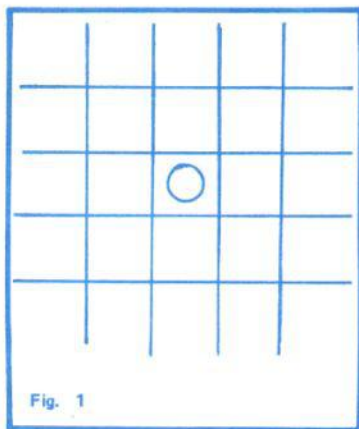



Fig. 1

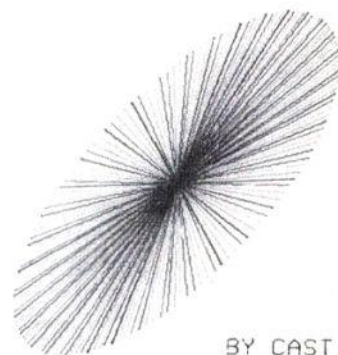
come griglia (Universo) le 25 righe e le 40 colonne dello schermo (25 × 40 = 1000 celle distinte). Essendo celle binarie (vuote o piene, vive o morte), ci danno la bellezza di 2^{1000} configurazioni iniziali differenti. C'è ne da fare osservazioni per anni!

Nella figura 2 è riportato il diagramma di flusso della simulazione. La cosa più importante da tener presente è che non bisogna cambiare lo stato di una cella, se prima il suo valore non è stato controllato da tutte le celle cui è adiacente. (Ogni cella appartiene simultaneamente all'intorno delle 8 celle adiacenti). Sono quindi necessarie due matrici. Una che chiamiamo Generazione Presente (G.P.), riportata sullo schermo. Un'altra che chiamiamo Generazione Futura (G.F.) che è di transito. Infatti, dalla G.P., di ogni cella si calcola il successivo stato, secondo le regole di Conway. Si pone quindi il risultato in G.F.. Esauriti i controlli si trasferisce G.F. in G.P. e si ricomincia daccapo. Si ricordi che ogni cella è adiacente ad altre otto, e quindi viene controllata 9 volte. Una quando funge da cella centrale ed otto quando è un elemento costituente un intorno.

Il programma necessita allora di due puntatori. Il puntatore alla cella centrale (P.C.C.) ed il puntatore all'intorno (P.I.). Il primo per permettere di passare in rassegna la matrice, il secondo per il controllo degli intorni. Infine c'è bisogno di un contatore che rilevi il numero di cellule vive per ogni intorno.

L'uso del linguaggio macchina di conseguenza, è stato necessario, in quanto la velocità di calcolo è un requisito indispensabile per dare interesse alle osservazioni. Una prima versione Basic del programma impiegava quasi due minuti nel computo di una generazione, mentre adesso, nello stesso tempo, se ne calcolano 900 circa. Si è reso quindi necessario l'introduzione di un "pause". Tenendo premuto il tasto RUN/STOP, l'elaborazione si ferma fino a che questo non viene rilasciato.

La prima generazione viene "creata" casualmente. È possibile variare la pro-

babilità di vita, cioè la densità di celle occupate alla prima generazione, cambiando il valore di test della variabile "C" alla riga 260. Si tenga presente che aumentando quel numero aumenta la densità di cellule vive.

Valori superiori a 0.85 inferiori a 0.15 portano spesso alla estinzione completa della colonia.

Potete anche "creare" voi stessi una particolare configurazione iniziale. Spostatevi per lo schermo con gli usuali tasti (CRS) e date vita ad una cellula digitando "SHIFT Q" (non preoccupatevi dei caratteri eventualmente presenti che vengono interpretati come siti vuoti). Quando avete finito digitate "RUN 290".

Ho notato che esistono versioni del "sessantaquattro" che richiedono la preventiva importazione del colore di un carattere, quando questo viene immesso direttamente nella memoria video, affinché sia visibile. Coloro che sono in possesso di questa versione devono digitare "RUN 295". In figura 4 sono riportati alcuni organismi stabili che spesso si possono osservare. In figura 5 vi sono degli interessantissimi "Pulsar". In figura 6 si può vedere lo straordinario "vagante". Provate a digitarlo così come è riportato e guardate che cosa succede.

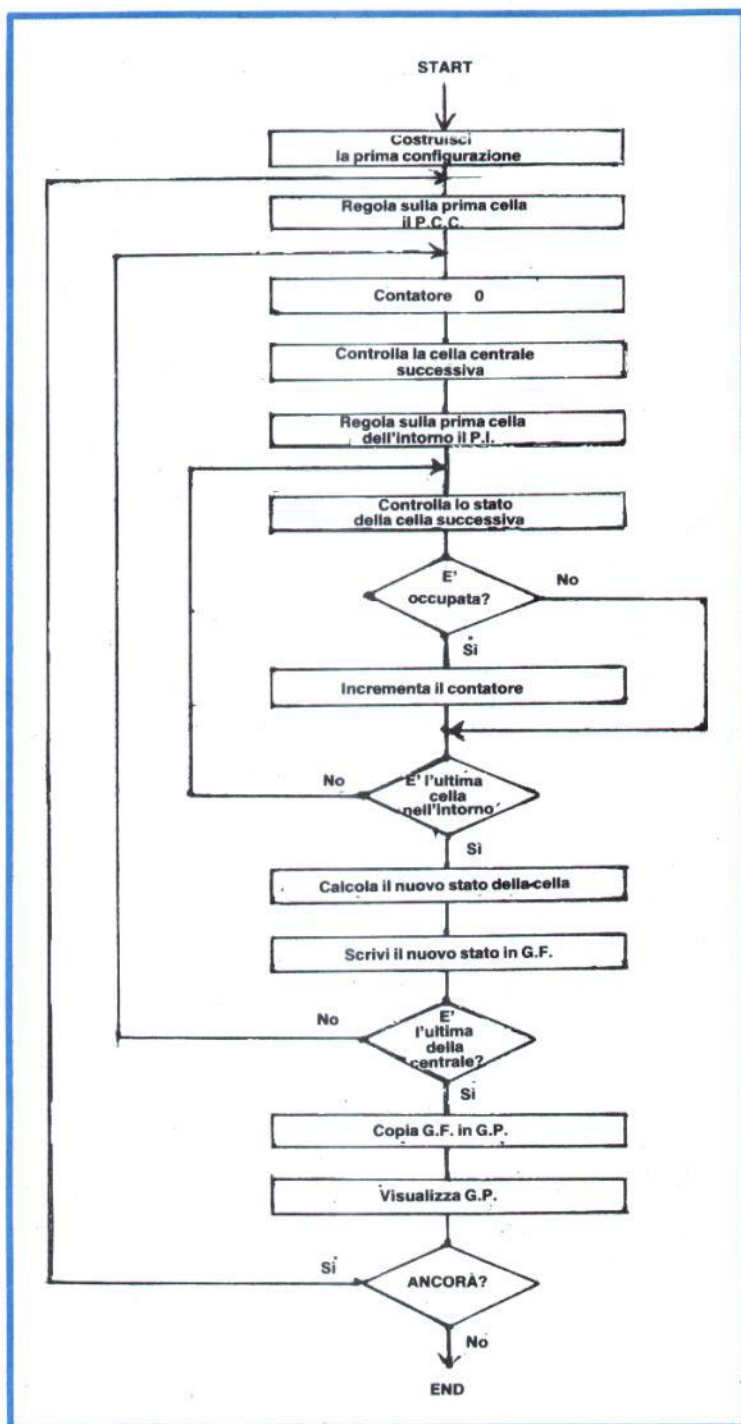
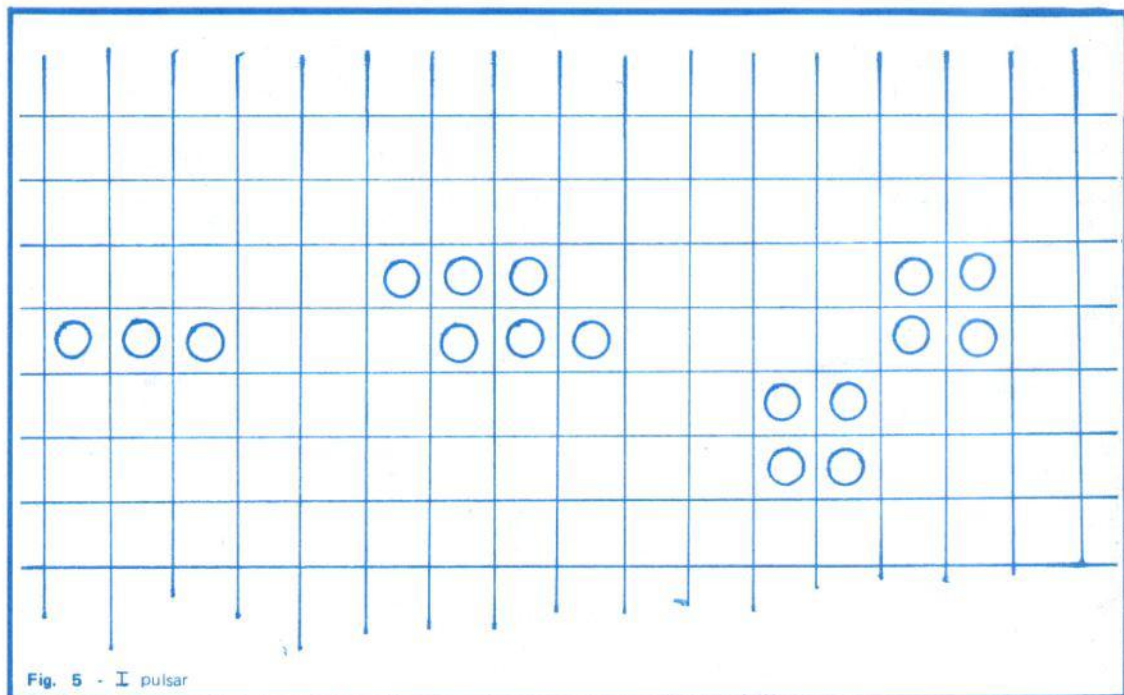
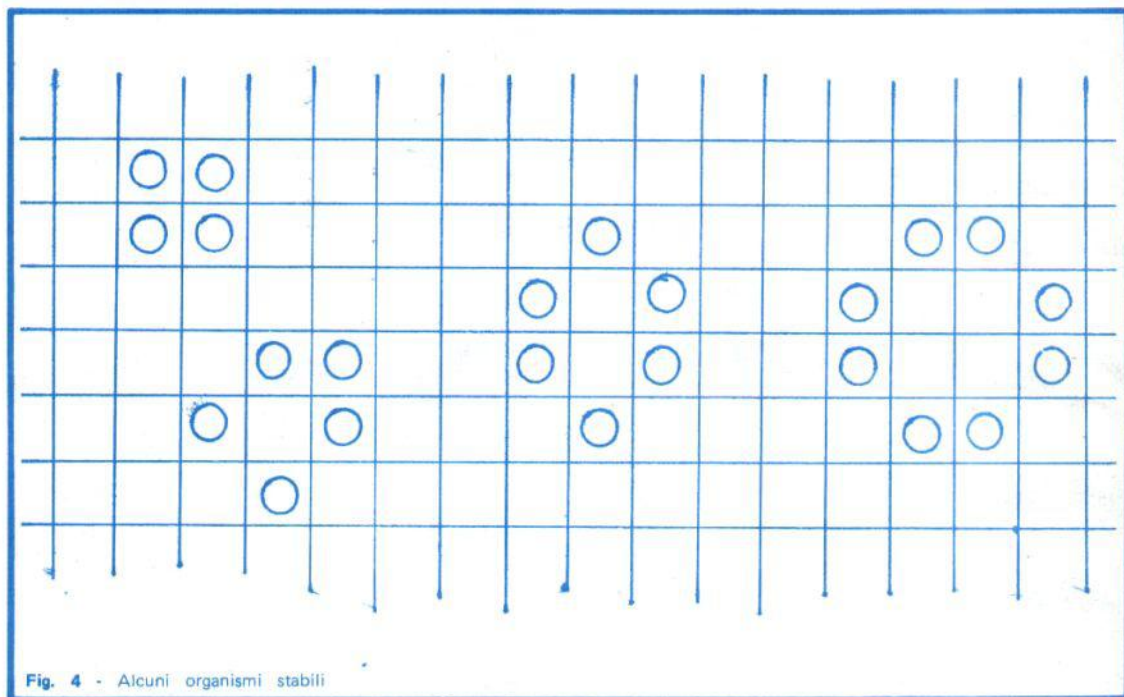
Si sappia inoltre che il bordo sinistro dello schermo è contiguo a quello destro, così come quello superiore lo è a quello inferiore. Abbiamo quindi un "universo" finito, ma privo di confini. Per ultima cosa voglio illustrare lo scopo delle linee 220-230.

Il programma in linguaggio macchina viene caricato da Basic alle linee 160-190, e bisogna attendere parecchi secondi, cosa che può rilevarsi molto fastidiosa quando si intende far girare il programma più di una volta consecutivamente.

Le linee 220-230 hanno pertanto la funzione di introdurre, *dopo il primo RUN*, una istruzione di salto per evitare di caricare più di una volta il programma in linguaggio macchina, ormai già presente in memoria.

Quindi, attenzione, quando salvate il

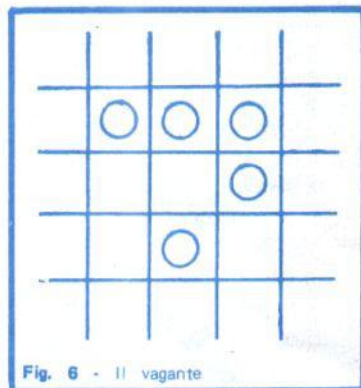
Fig. 4 - Alcuni organismi stabili

Fig. 5 - I pulsar

```
1350 DATA162,0,160,0
1360 DATA189,0,5,201,81,208,1,200
1370 DATA189,1,5,201,81,208,1,200
1380 DATA189,2,5,201,81,208,1,200
1390 DATA189,40,5,201,81,208,1,200
1400 DATA189,42,5,201,81,208,1,200
1410 DATA189,80,5,201,81,208,1,200
1420 DATA189,81,5,201,81,208,1,200
1430 DATA189,82,5,201,81,208,1,200
1440 DATA192,3,240,28
1450 DATA189,41,5,201,81,208,5,200,192,3,240,16,169,32
1460 DATA157,0,201
1470 DATA232,224,0,208,3
1480 DATA76,232,192
1490 DATA76,118,192
1500 DATA169,81
1510 DATA157,0,201
1520 DATA232,224,0,208,3
1530 DATA76,232,192
1540 DATA76,118,192
1550 DATA162,0,160,0
1560 DATA189,0,6,201,81,208,1,200
1570 DATA189,1,6,201,81,208,1,200
1580 DATA189,2,6,201,81,208,1,200
1590 DATA189,40,6,201,81,208,1,200
1600 DATA189,42,6,201,81,208,1,200
1610 DATA189,80,6,201,81,208,1,200
1620 DATA189,81,6,201,81,208,1,200
1630 DATA189,82,6,201,81,208,1,200
1640 DATA192,3,240,28
1650 DATA189,41,6,201,81,208,5,200,192,3,240,16,169,32
1660 DATA157,0,202
1670 DATA232,224,0,208,3
1680 DATA76,92,193
1690 DATA76,234,192
1700 DATA169,81
1710 DATA157,0,202
1720 DATA232,224,0,208,3
1730 DATA76,92,193
1740 DATA76,234,192
1750 DATA162,0,160,0
1760 DATA189,0,7,201,81,208,1,200
1770 DATA189,1,7,201,81,208,1,200
1780 DATA189,2,7,201,81,208,1,200
1790 DATA189,40,7,201,81,208,1,200
1800 DATA189,42,7,201,81,208,1,200
1810 DATA189,80,7,201,81,208,1,200
1820 DATA189,81,7,201,81,208,1,200
1830 DATA189,82,7,201,81,208,1,200
1840 DATA192,3,240,28
1850 DATA189,41,7,201,81,208,5,200,192,3,240,16,169,32
1860 DATA157,0,203
```

programma: assicuratevi che non sia presente la linea 155. Dopo il primo RUN essa comparirà da sola, automaticamente e vi permetterà di risparmiare tempo per fare un'altra "osservazione". Per uscire dal programma bisogna premere "RUN/STOP" e "RESTORE". Se volete sapere il numero di generazioni calcolate digitate RUN 3333.

Antonio Visconti

Fig. 6 - Il vagante

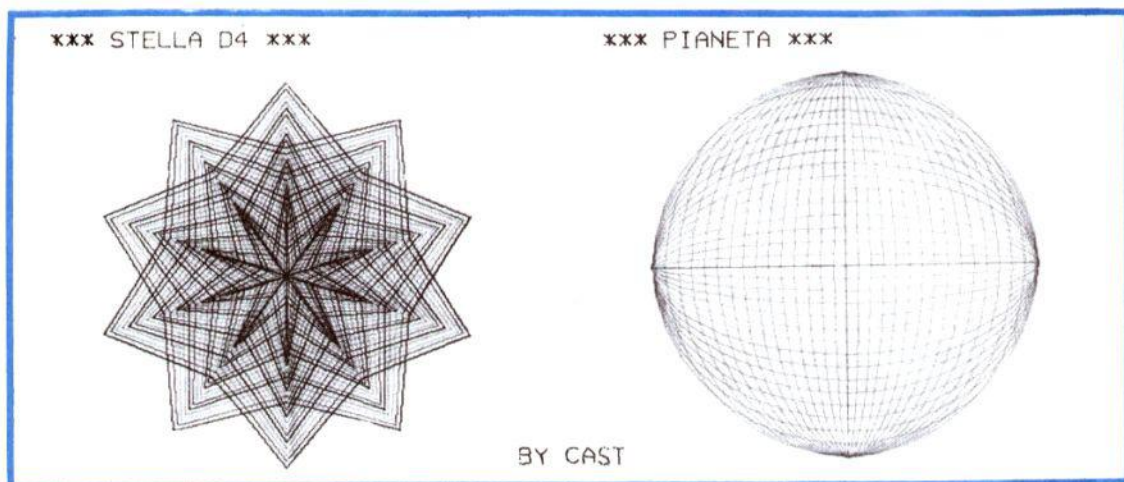
```
0 REM**********************************
1 REM**        COMMODORE  64          **
2 REM**            VITA               **
3 REM**             BY                **
4 REM**      ANTONIO     VISCONTI     **
8 REM**********************************
157 PRINTCHR$(147):POKE53280,0:POKE53281,7
158 PRINT"ATTENDERE, PREGO..."
160 I=0:UU=0
170 READQ:IFQ<0THEN220
180 POKE49152+I,Q:UU=UU+Q
190 I=I+1:GOTO170
220 IFUU<>95325THENPRINT"ERRORE NEI DATA":END
221 POKE631,13:POKE632,13:POKE198,2
225 PRINT"■◣":PRINT:PRINT
230 PRINT"155FORI=1TO1500:NEXT:GOTO240":PRINT"RUN240":PRINT"□□□□□":END
240 PRINT"◣":POKE53269,0:POKE53280,0:POKE53281,0
250 FOR K=0 TO 999
260 C=RND(1):IFC<.35THENA=81:GOTO280
270 A=32
280 POKE55296+K,14:POKE1024+K,A:NEXT
290 POKE53280,0:POKE53281,0:POKE822,0:POKE823,0:SYS49152
295 FORI=55296TO56295:POKEI,14:NEXT
296 GOTO290
1150 DATA 162,0,160,0
1160 DATA189,0,4,201,81,208,1,200
1170 DATA189,1,4,201,81,208,1,200
1180 DATA189,2,4,201,81,208,1,200
1190 DATA189,40,4,201,81,208,1,200
1200 DATA189,42,4,201,81,208,1,200
1210 DATA189,80,4,201,81,208,1,200
1220 DATA189,81,4,201,81,208,1,200
1230 DATA189,82,4,201,81,208,1,200
1240 DATA192,3,240,28
1250 DATA189,41,4,201,81,208,5,200,192,3,240,16,169,32
1260 DATA157,0,200
1270 DATA232,224,0,208,3
1280 DATA76,116,192
1290 DATA76,2,192
1300 DATA169,81
1310 DATA157,0,200
1320 DATA232,224,0,208,3
1330 DATA76,116,192
1340 DATA76,2,192
```

```
1880 DATA232,224,150,208,3,76,208,193,76
1885 DATA94,193,169,81,157,0,203,232,224,150
1890 DATA208,3,76,208,193,76,94,193
1900 DATA162,0,160,0
1910 DATA189,150,7,201,81,208,1,200,224,81,48,6
1920 DATA189,175,3,76,233,193
1930 DATA189,151,7,201,81,208,1,200,224,80,48,6
1940 DATA189,176,3,76,251,193,189,152,7
1950 DATA201,81,208,1,200,224,42,48,6
1960 DATA189,214,3,76,13,194,189,190,7
1970 DATA201,81,208,1,200,224,40,48,6
1980 DATA189,216,3,76,31,194,189,192,7
1990 DATA201,81,208,1,200,224,2,48,6
2000 DATA189,254,3,76,49,194,189,230,7
2010 DATA201,81,208,1,200,224,1,48,6
2020 DATA189,255,3,76,67,194,189,231,7
2030 DATA201,81,208,1,200
2040 DATA189,0,4,201,81,208,1,200,192,3
2050 DATA240,38,224,41,48,6
2060 DATA189,215,3,76,97,194,189,191,7
2070 DATA201,81,208,5,200,192,3,240,16,169,32
2080 DATA157,150,203,232,224,82,208,3
2090 DATA76,138,194,76,210,193
2100 DATA169,81,157,150,203
2110 DATA232,224,82,208,3
2120 DATA76,138,194,76,210,193
2140 DATA162,0
2150 DATA189,0,200
2160 DATA157,41,4,232,224,0,208,3
2170 DATA76,157,194
2180 DATA76,140,194
2200 DATA162,0
2210 DATA189,0,201
2220 DATA157,41,5,232,224,0,208,3
2230 DATA76,176,194
2240 DATA76,159,194
2250 DATA162,0
2260 DATA189,0,202
2270 DATA157,41,6,232,224,0,208,3
2280 DATA76,195,194
2290 DATA76,178,194
2300 DATA162,0,189,0,203,157,41,7,232
2305 DATA224,150,208,3,76,214,194,76,197,194
2310 DATA162,0,189,150,203,157,191,7,189,191,203
2320 DATA157,0,4,232,224,41,208,3,24,144,3,76,216,194
2330 DATA32,234,255,32,225,255,240,11,238,54,3,208,3,238,55,3,76,0,192
2340 DATA32,225,255,240,11,238,54,3,208
2345 DATA3,238,55,3,76,0,192,76,2,195,-9,-9
3333 CC=PEEK(822)+PEEK(823)*256:PRINT"IL NUMERO DI GENERAZIONI "
3334 PRINT"CALCOLATE E' STATO "CC:END

READY.
```

SAVE

Scopo del gioco guidare l'omino che deve attraversare lo schermo da sinistra a destra. Arrivato alla fanciulla (a sinistra dello schermo) per far riapparire l'omino a destra basta premere uno dei tasti di controllo consentiti.

Ostacoli: bisogna far saltare l'omino di roccia in roccia altrimenti cade nei crepacci perdendo una delle sue 3 vite. Ogni tanto dal cielo piovono dei macigni che, raggiungendo inoltre l'uccello tro-dos vola ininterrottamente nel cielo. Attenti a non urtarlo. Ultimo ostacolo è il brontosauro Jerry (situato nell'ultimo crepaccio): saltategli sopra quando tira fuori dal crepaccio la testa altrimenti cascate giù.

Il gioco termina quando perdete le tre vite a disposizione.

Andrea Campione

```
100 REM        S A V E
110 REM    PER VIC 20 INESPANSO
120 REM
130 REM   DI ANDREA CAMPIONE
170 :
171 PRINT"USA I TASTI:
172 PRINT"Z (SALTO)"
173 PRINT"< (SINISTRA)"
174 PRINT"> (DESTRA)"
175 PRINT"PREMI UN TASTO"
176 GETA$:IFA$=""THEN176
180 POKE36879,110:PRINT"":POKE36878,15
190 POKE36869,255:FORI=7168TO8185:POKEI,0:NEXT:POKE37154,127
200 FORZ=7176TO7183:READA:POKEZ,A:NEXT
210 FORZ=7184TO7191:READB:POKEZ,B:NEXT
```

```
220 FORZ=7192TO7199:READC:POKEZ,C:NEXT
230 FORZ=7200TO7207:READD:POKEZ,D:NEXT
240 FORZ=7208TO7215:READE:POKEZ,E:NEXT
250 FORZ=7216TO7223:READF:POKEZ,F:NEXT
260 FORZ=7224TO7231:READG:POKEZ,G:NEXT
270 FORZ=7232TO7239:READH:POKEZ,H:NEXT
280 POKE36879,111
290 PRINT"AAAA F AAA A A AA AAA"
300 PRINT"A AA G AA A A AAA"
310 PRINT"AAAA G A A A AA A"
320 GOSUB1060
330 IFPEEK(W+1)=2 OR PEEK(W+1)=3THEN GOSUB 1200
340 IFV=0THENGOTO890
350 K=INT(50*RND(1))
360 IFK>40THENGOSUB1020
370 IFK<40THENGOTO380
380 J=INT(50*RND(1))
390 IFPEEK(R-21)=2ORPEEK(R-21)=3THENV=V-1:GOSUB790
400 IFPEEK(R+23)=2ORPEEK(R+23)=3THENV=V-1:GOSUB790
410 IFPEEK(R+23)=1THENPOKER,0:R=7900
420 GOSUB530
430 IFR+23>8185THENPOKER,0:R=7900
440 IFR-21<7680THENPOKER,0:R=7900
450 IFJ>40THENPOKER,0:POKER-21,5:R=R-21:GOSUB760
460 IFJ<40THENPOKER,0:POKER+23,5:R=R+23:GOSUB760
470 IFW+1>8096THENPOKEW,0:W=8009
480 POKEW,0:POKEW+1,4:W=W+1:GOSUB860
490 FORX=1TO2:NEXT
500 POKE8076,8
510 GOSUB530
520 GOTO330
530 IFPEEK(197)=37THENGOTO580
540 IFPEEK(197)=29THENGOTO620
550 IFPEEK(197)=33THENGOTO660
560 IFPEEK(Q+22)=0THENV=V-1:GOSUB790:GOSUB970:RETURN
570 RETURN
580 IFQ+1>8096THENQ=Q-1
590 POKEQ,0:POKEQ+1,2:Q=Q+1
600 IFPEEK(Q+22)<>1THENV=V-1:GOSUB970:GOSUB790:Q=8096:RETURN
610 RETURN
620 IFPEEK(Q-1)=8THEN GOSUB 1220:RETURN
630 POKEQ,0:POKEQ-1,2:Q=Q-1
640 IFPEEK(Q+22)<>1THENV=V-1:GOSUB790:GOSUB970:POKEQ,0:Q=8096:RETURN
650 RETURN
660 POKEQ,0:POKEQ-23,3:Q=Q-23
670 FORX=1TO50:NEXTX
680 IFPEEK(Q+22)=4THENP=P+500:PRINT"PETER="P"VITE="V
690 POKEQ,0:POKEQ+21,3:Q=Q+21
700 IFPEEK(Q+22)=6THENRETURN
710 IFPEEK(Q+22)<>1THENV=V-1:GOSUB790:GOSUB970:Q=8096:RETURN
720 RETURN
730 POKE36876,252
```

```
740 POKE36876,0
750 RETURN
760 POKE36874,128
770 POKE36874,0
780 RETURN
790 PRINT"PETER="P"VITE="V
800 POKE36877,200
810 FORX=1TO100
820 NEXT
830 POKE36877,0
840 POKEQ,0:POKEQ,2
850 RETURN
860 POKE36875,200
870 POKE36875,0
880 RETURN
890 POKE37154,255:POKE36869,240:POKE36879,27
900 PRINT"PECCATO!HAI PERSO LE "
910 PRINT"TUE TRE VITE!"
920 PRINT"PUNTEGGIO="P
930 POKE198,0:PRINT"GIOCHI ANCORA (S/N)":INPUTA$
940 IFA$="N"THENEND
950 IFA$="S"THENPOKE36879,110:POKE36878,15:RUN180
960 IFA$<>"S"ORA$<>"N"THEN930
970 FORU=1TO4
980 POKEQ,0:POKEQ+22,3:Q=Q+22
990 FORX=1TO100:NEXT
1000 NEXT
1010 POKEQ,0:RETURN
1020 IFPEEK(T)<>6THENGOTO1040
1030 IFPEEK(T)=6THENGOTO1050
1040 POKEY,7:POKEY-22,6:GOSUB730:RETURN
1050 POKET,0:POKET+22,6:GOSUB730:RETURN
1060 V=3
1070 P=0:PRINT"PETER="P"VITE="V
1080 Q=8096:W=8009:R=7900
1090 Y=8125:T=8103
1100 POKEQ,2:POKEW,4:POKET,6:POKEY,7:POKE8076,8
1110 RETURN
1120 DATA255,126,38,110,28,62,127,255
1130 DATA76,76,124,12,12,20,36,204
1140 DATA155,126,24,24,36,66,129,0
1150 DATA138,79,44,252,60,16,12,0
1160 DATA24,60,66,149,217,70,60,24
1170 DATA60,90,60,66,66,126,60,0
1180 DATA60,0,60,0,60,0,60,60
1190 DATA240,160,224,255,255,224,224,224
1200 V=V-1:PRINT"PETER="P"VITE="V:GOSUB790
1210 POKEQ,0:Q=8096:RETURN
1220 P=P+1000:PRINT"PETER="P"VITE="V:POKEQ,0
1230 Q=8096:GOSUB790:RETURN

READY.
```

PRIME NOTE

Scopo del programma è quello di imparare, giocando, i primi rudimenti della notazione musicale.

Sullo schermo appare un pentagramma con la chiave di violino. Si può scegliere di giocare a leggere le note scrivendone il nome. (Ricordare di fare seguire al nome della nota il numero 1 se la nota è nell'ottava bassa, 2 se in quella alta - Es. Sol2 Do1 ecc.) oppure scriverle portando il cursore con le frecce nel rigo o nello spazio opportuno e premendo return.

Sullo schermo in alto a destra appare il punteggio raggiunto in base all'esattezza delle risposte e alla velocità.

In alto a sinistra sono indicati i punteggi massimi sia in lettura che in scrittura stabiliti nelle partite precedenti.

A ogni risposta esatta la nota viene suonata con dolcezza: se invece la risposta è sbagliata l'errore viene corretto e viene emesso un suono poco gradevole.

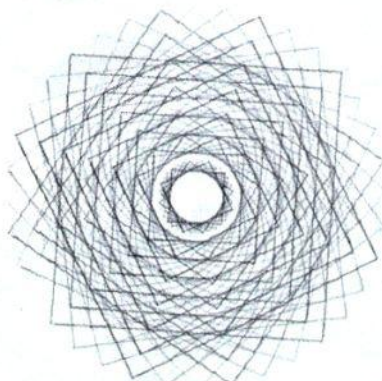
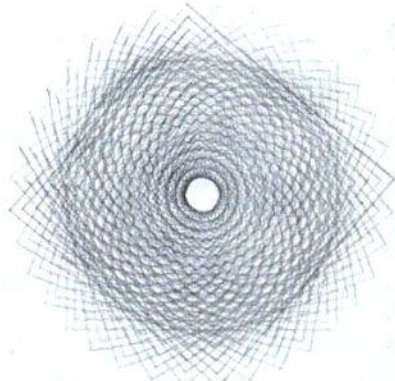
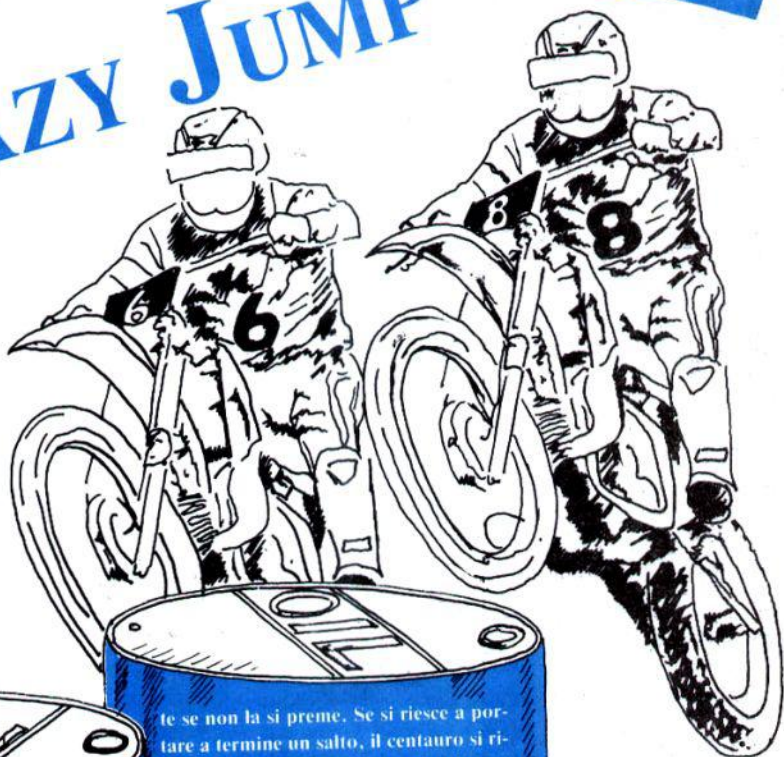
A fine partita appare un giudizio: da "molto scarso" a "perfetto" e le 23 note su cui si è giocato vengono suonate in successione in modo gradevole se si è raggiunta almeno la "sufficienza" sgradevole in caso contrario.

Maria Serena Borgia
(di anni 13)

```
100 REM ******************************
110 REM *   P R I M E   N O T E      *
120 REM *   IMPARARE IL PENTAGRAMMA   *
130 REM * GIOCANDO SUL COMMODORE 64 *
140 REM *    MARIA SERENA BORGIA      *
160 REM *           BOLOGNA           *
180 REM ******************************
190 GOSUB300
200 GOSUB220:IFA$="S"THEN200
210 GOSUB430:END
220 GOSUB460
230 FORI=0TO23:GOSUB250:NEXT:GOSUB660
240 RETURN
250 GOSUB900
260 IFA$="L"THENGOSUB950
270 IFA$="S"THENGOSUB1100
280 GOSUB1250
290 RETURN
300 REM INIZIO PROGRAMMA
310 V=53248:POKEV+32,6
320 PRINT"          P R I M E   N O T E"
330 PRINT"          ======================="
332 DIMT1$(12):DIMT2(12,3):DIMT3(24)
334 FORN=0TO11:READT1$(N):NEXT
```

```
336 FORN=0TO11:READT2(N,0),T2(N,1),T2(N,2):NEXT
340 POKE2040,13
350 FORN=0TO62:READQ:POKE832+N,Q:NEXT
360 POKEV,30:POKEV+1,100
380 POKEV+39,0:POKEV+23,3:POKEV+29,3
400 RL=0:RS=0:S=54272:POKES+24,15
410 POKES+5,0:POKES+6,249
420 RETURN
430 REM FINE PROGRAMMA
440 PRINT"▣"
450 RETURN
460 REM INIZIO PARTITA
470 PU=0
480 PRINT"▣RECORD LETTURA    ";RL;
490 PRINTTAB(35);"0"
500 PRINT"RECORD SCRITTURA ";RS;"■"
510 FORCR=1TO5:PRINT
520 FORN=1TO39:PRINT"—";:NEXT:PRINT
530 NEXT:POKEV+21,1
540 PRINT"■■■■PROVA DI LETTURA "
550 PRINT"O DI SCRITTURA ■(L/S)▣"
560 GETA$:IFA$=""THEN560
570 IF(A$<>"L"ANDA$<>"S")THEN560
580 IFA$="S"THEN620
590 PRINT"□□SCRIVI CHE NOTA E'      "
600 PRINT"SEGUITA DA 1 SE OTTAVA BASSA"
605 PRINT"              2 SE OTTAVA ALTA"
606 PRINT"E POI PREMI ■RETURN■■"
610 GOTO640
620 PRINT"□□DOVE VA SCRITTA QUESTA NOTA?"
630 PRINT"SPOSTA IL CURSORE E PREMI ■RETURN■"
635 PRINT"■ 1=OTTAVA BASSA   2=OTTAVA ALTA"
640 PRINT"■■PRESTO: ANCHE IL TEMPO CONTA.."
650 NP=99:I=0
655 RETURN
660 REM FINE PARTITA
670 IF(A$="S"ANDPU>RS)THENRS=PU
680 IF(A$="L"ANDPU>RL)THENRL=PU
690 PRINT"■";:FORN=1TO21:PRINT"■";:NEXT:PRINT"■";
692 PRINT"                                    "
694 PRINT"□       ";
700 IFPU<100THENPRINT"MOLTO MALE":GOTO770
710 IFPU<200THENPRINT"MALE       ":GOTO770
720 IFPU<300THENPRINT"SCARSO     ":GOTO770
730 IFPU<500THENPRINT"SUFFICIENTE":GOTO770
740 IFPU<700THENPRINT"BENE       ":GOTO770
750 IFPU<1000THENPRINT"OTTIMO     ":GOTO770
760 PRINT"PERFETTO!!"
770 PRINT"▣":POKES,0:POKES+1,0
772 PRINT"▣":POKES,0:POKES+1,0
780 IFPU<300THENPOKES+4,129
790 IFPU>299THENPOKES+4,17
```

```
800 FORI=0TO23
802 N=T3(I):PRINT"■";
803 FORQ=1TO14:PRINT"■";:NEXT
804 PRINT"                                        ":PRINT"□";
805 PRINTTAB(6+I);"■";T1$(N);"■"
810 POKES,T2(N,1):POKES+1,T2(N,0)
812 PRINT"■";:FORQ=1TOT2(N,2):PRINT"■";:NEXTQ
813 IFN=0THENPRINTTAB(6+I);"■─■●■─":GOTO820
814 PRINTTAB(6+I);"■●■"
820 FORN=0TO400:NEXTN:NEXTI
825 POKES+4,0:FORN=0TO1000:NEXT
830 POKEV+21,0
840 PRINT"■■■■■";SPC(11);"■ALTRA PARTITA (S/N)■"
850 GETA$:IFA$=""THEN850
860 IFA$<>"S"ANDA$<>"N"THEN850
900 REM INIZIO MOSSA
910 NO=INT(RND(1)*12)
920 IFNO=NPTHEN910
930 NP=NO:TE=TI:T3(I)=NO
940 RETURN
950 REM LETTURA
960 PRINT"■";
970 FORN=1TOT2(NO,2):PRINT"■";:NEXT
980 IFNO=0THENPRINTTAB(6+I);"■─■●■─■":GOTO1000
990 PRINTTAB(6+I);"■●■"
1000 IFI=0THEN1009
1002 PRINT"■";:Q=T3(I-1)
1003 FORN=1TOT2(Q,2):PRINT"■";:NEXT
1005 IFQ=0THENPRINTTAB(6+I);"■●■":GOTO1009
1007 PRINTTAB(5+I);"■●■"
1009 PRINT"■";
1010 FORN=1TO14:PRINT"■";:NEXT
1015 FORN=0TO6+I+5:PRINT"  ";:NEXT
1020 PRINT"■■■■■■■■";:INPUTNO$
1030 IFNO$=T1$(NO)THENOK$="SI":GOTO1050
1040 OK$="NO"
1050 NO$="  "
1055 RETURN
1100 REM SCRITTURA
1110 PRINT"■";
1120 FORN=1TO14:PRINT"■";:NEXT
1125 FORN=0TO6+I:PRINT"  ";:NEXT
1130 PRINT"■■■■";T1$(NO);
1140 INPUTNO$
1150 IFPEEK(214)=T2(NO,2)+1THENOK$="SI":GOTO1170
1160 OK$="NO"
1170 PRINT"■";
1180 FORN=1TOT2(NO,2):PRINT"■";:NEXT
1190 IFNO=0THENPRINTTAB(6+I);"■─●─■":GOTO1210
1200 PRINTTAB(6+I);"■●■"
1210 RETURN
1250 REM FINE MOSSA
```

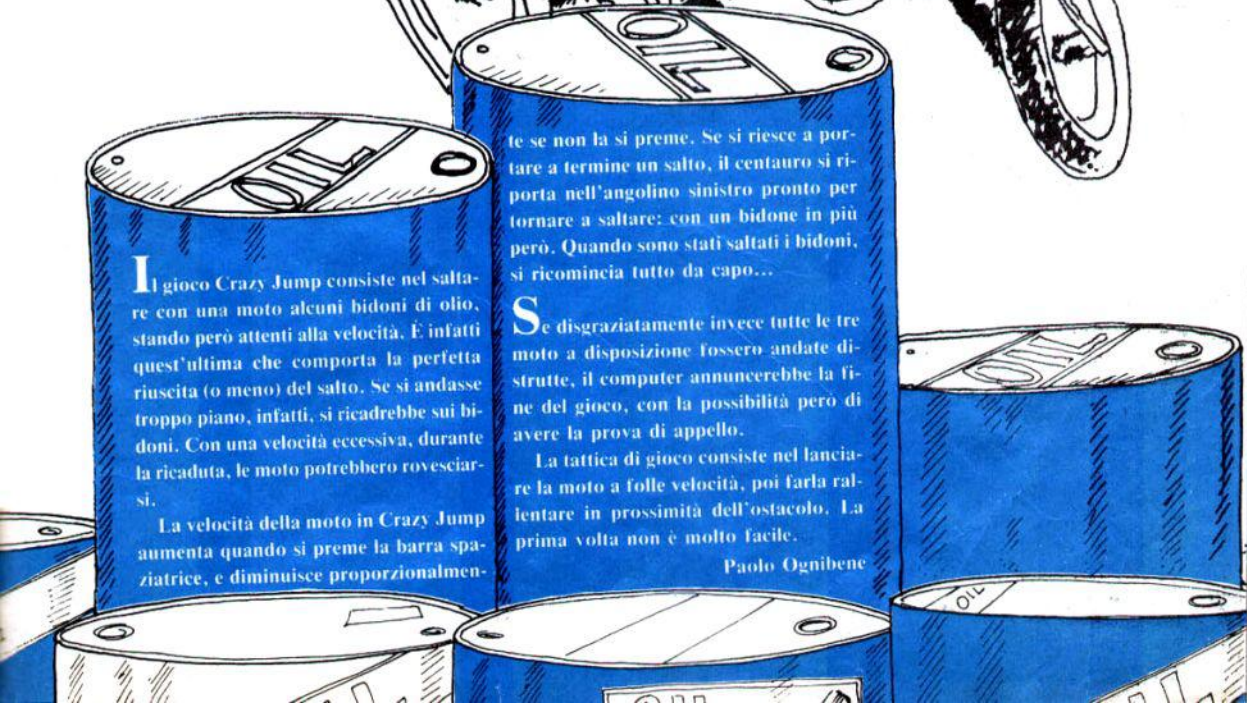
```
1260 IFOK$="NO"THEN1300
1270 TE=INT(50-(TI-TE)/60)
1280 IFTE<1THEN1300
1290 PU=PU+TE:PRINT"■■";TAB(35);PU;"■"
1300 PRINT"■";:FORN=1TO21:PRINT"■";:NEXT
1310 IFOK$="NO"THEN1340
1320 PRINT"■        B E N E  ■           "
1330 GOTO1360
1340 PRINT"■ A H I !! ■";
1350 IFA$="L"THENPRINT"E' UN ■";T1$(NO);" ■"
1355 IFA$="S"THENPRINT"                    "
1360 POKES,T2(NO,1):POKES+1,T2(NO,0)
1370 IFOK$="SI"THENPOKES+4,33:GOTO1390
1380 POKES+4,129
1390 FORN=0TO100:NEXT
1392 IFOK$="SI"THENPOKES+4,32
1394 IFOK$="NO"THENPOKES+4,128
1400 RETURN
1410 REM NOME NOTE
1420 DATA DO1,RE1,MI1,FA1,SOL1,LA1,SI1
1430 DATA DO2,RE2,MI2,FA2,SOL2
1440 REM FREQUENZE E POSIZIONI NOTE
1450 DATA 8,180,13,9,196,12,10,247,11
1460 DATA 11,158,10,13,10,9,14,162,8
1470 DATA 16,109,7,17,103,6,19,137,5
1480 DATA 21,237,4,23,59,3,26,20,2
1490 REM SPRITES CHIAVE DI VIOLINO
1500 DATA 0,6,0,0,9,0,0,9,0,0,9,0
1510 DATA 0,12,0,0,10,0,0,14,0,0,28,0
1520 DATA 0,56,0,0,104,0,0,223,0,1,31,0
1530 DATA 1,41,128,1,41,0,1,153,0
1540 DATA 0,254,0,0,4,0,0,4,0,0,100,0
1550 DATA 0,100,0,0,120,0
READY.
```

CRAZY JUMP

Il gioco Crazy Jump consiste nel saltare con una moto alcuni bidoni di olio, stando però attenti alla velocità. È infatti quest'ultima che comporta la perfetta riuscita (o meno) del salto. Se si andasse troppo piano, infatti, si ricadrebbe sui bidoni. Con una velocità eccessiva, durante la ricaduta, le moto potrebbero rovesciarsi.

La velocità della moto in Crazy Jump aumenta quando si preme la barra spaziatrice, e diminuisce proporzionalmente se non la si preme. Se si riesce a portare a termine un salto, il centauro si riporta nell'angolino sinistro pronto per tornare a saltare: con un bidone in più però. Quando sono stati saltati i bidoni, si ricomincia tutto da capo...

Se disgraziatamente invece tutte le tre moto a disposizione fossero andate distrutte, il computer annuncerebbe la fine del gioco, con la possibilità però di avere la prova di appello.

La tattica di gioco consiste nel lanciare la moto a folle velocità, poi farla rallentare in prossimità dell'ostacolo. La prima volta non è molto facile.

Paolo Ognibene

```
100 REM ***              CRAZY JUMP               ***
110 REM ***   GIOCO PER VIC 20 INESPANSO          ***
120 REM ***          DI PAOLO OGNIBENE            ***
150 :
160 PRINT"■":POKE36879,30
170 POKE36869,255:Y=7790:TT=50:D=30720:S=7168:PX=3:OX=3
180 FORI=STOS+10*7+9:READA:POKEI,A:NEXT
190 PRINT"■■■■■■VIC CRAZY JUMP■"
200 PRINT"■■VELOC.:0■":PRINT"■■■■■■■■■■■■■■■■■"PX" "OX"■"
210 GOSUB410
220 POKEY,0:POKEY+D,0
230 GETA$:IFA$<>" "THENAT=0:GOTO230
240 POKEY+D-1,1:POKEY,0
250 IFPEEK(8101)=0THEN480
260 GETA$:IFA$=" "THENTT=TT*.95:GOTO280
270 TT=TT*1.05
280 IFTT<5THENTT=5
290 FORT=0TOTT:NEXT
300 VE=(INT((1/TT)*100)-1)*10
310 PRINT"■■■■VELOC.:      ■"
320 PRINT"■■■■VELOC."VE" ■"
330 POKEY+1,2:POKEY,1:POKEY+1+D,0
340 FORT=0TOTT:NEXT
350 Y=Y+1
360 IFY=7811THENPOKEY+D,1:POKEY+D-1,1:Y=7899
370 IFY=7921THENPOKEY+D,1:POKEY+D-1,1:Y=8097
380 IFTT>50THENAT=AT+1:TT=50
390 IFAT=5THEN230
400 GOTO240
410 FORCE=7812TO7833:POKECE,8:POKECE+D,5:NEXT
420 FORCE=7922TO7943:POKECE,8:POKECE+D,5:NEXT
430 FORCE=8120TO8141:POKECE,8:POKECE+D,5:NEXT
440 POKE8102,5:POKE8102+D,0
450 FOROS=8103TO8102+OX:POKEOS,6:POKEOS+D,0:NEXTOS
460 POKEOS,7:POKEOS+D,0
470 RETURN
480 IFVE>=80ANDVE<=210THEN500
490 IFVE<80THEN890
500 POKEY+D,1:POKEY+1,5:Y=Y+1
510 FORSS=1TO2:Y=Y-21:FORT=0TOTT:NEXTT:POKE8102,5
520 POKE8102+D,0:POKEY,3:POKEY+D+21,1:POKEY+D,0:NEXTSS
530 IFVE>=80ANDVE<=110THEN560
540 IFVE>=111ANDVE<=190THEN610
550 IFVE>=191ANDVE<=210THEN620
560 FORGY=1TOVE/10-8
570 Y=Y+1
580 POKEY,3:POKEY+D-1,1:POKEY+D,0
590 FORT=0TOTT*4:NEXTT:NEXTGY
600 GOTO640
610 FORT=0TOTT*4:NEXTT:Y=Y-21:POKEY,3:POKEY+D+21,1:POKEY+D,0:GOTO560
```

```
620 FORPS=1TO2:FORT=0TOTT*4:NEXTT:Y=Y-21:POKEY,3
630 POKEY+D+21,1:POKEY+D,0:NEXTPS:GOTO560
640 Y=Y+23
650 IFY=8098ORY=8120ORY=8121ORY=8143ORY=8142THENPOKEY-23+D,1:GOTO950
660 IFY>=8120ANDY<=8141THEN680
670 FORT=0TOTT*4:NEXTT:POKEY,3:POKEY+D-23,1:POKEY+D,0:GOTO640
680 IFOX=3ANDVE>=100ANDVE<=110THEN810
690 IFOX=4ANDVE>=110ANDVE<=120THEN810
700 IFOX=5ANDVE>=120ANDVE<=130THEN810
710 IFOX=6ANDVE>=120ANDVE<=130THEN810
720 IFOX=7ANDVE>=120ANDVE<=130THEN810
730 IFOX=8ANDVE>=130ANDVE<=140THEN810
740 IFOX=9ANDVE>=140ANDVE<=150THEN810
750 IFOX=10ANDVE>=150ANDVE<=160THEN810
760 IFOX=11ANDVE=160THEN810
770 Y=Y-23
780 FORWY=YTO8118:FORT=0TOTT*4:NEXTT:POKEWY-1+D,1
790 POKEWY,9:POKEWY+D,0:TT=TT+4:NEXTWY
800 GOTO950
810 Y=Y-23
820 FORWY=YTO8118:FORT=0TOTT*4:NEXTT:POKEY-1+D,1:POKEY,0
830 FORT=0TOTT*4:NEXTT:POKEY+1,2:POKEY,1
840 POKEY+1+D,0:Y=Y+1:TT=TT+2:NEXTWY:FORT=1TO700:NEXTT
850 OX=OX+1:Y=7790:TT=50:IFOX=12THEN870
860 GOTO190
870 PRINT"■■■■■■■■■ PERFETTO! ■"
880 PRINT"■■■■VINCI SEMPRE!■":FORT=0TO3000:NEXTT:RUN
890 POKEY+D,1:POKEY+1,5:Y=Y+1
900 FORSS=1TO2:Y=Y-21:FORT=0TOTT*2:NEXTT:POKE8102,5
910 POKE8102+D,0:POKEY,3:POKEY+D+21,1:POKEY+D,0:NEXTSS
920 FORSS=1TO2:FORT=0TOTT*6:NEXTT:POKEY+22,0
930 POKEY+D,1:POKEY+D+22,0:NEXTSS
940 POKEY+22+D,1:POKEY+44,9
950 Y=7790:PX=PX-1:VE=0:TT=50:IFPX=0THEN970
960 FORT=0TO2500:NEXT:GOTO190
970 FORT=0TO800:NEXT:PRINT"■■  GIOCHI ANCORA (S/N) ■"
980 GETS$:IFS$=""THEN980
990 IFS$="S"THENRUN
1000 IFS$="N"THENSYS64802
1010 IFS$<>"S"ORS$<>"N"THEN980
1020 DATA0,4,24,62,52,90,173,66
1030 DATA0,0,1,3,3,5,10,4
1040 DATA0,64,128,224,64,160,208,32
1050 DATA0,32,122,101,122,72,160,64
1060 DATA0,32,122,101,123,79,175,95
1070 DATA0,0,0,1,3,7,15,31
1080 DATA0,0,0,127,77,93,73,127
1090 DATA0,0,0,64,96,112,120,124
1100 DATA255,255,0,255,0,0,255,0
1110 DATA0,66,181,94,44,60,12,8

READY.
```

FLIP!

Scopo del gioco: far scoppiare tutti i palloni visualizzati in alto sullo schermo con due omini che saltano ininterrottamente a turno su un trampolino.

Il gioco termina se un omino cade sull'altro o se non cade sul trampolino.

Andrea Campione

```
100 REM *         FLIP
110 REM * PER VIC 20 INESPANSO
120 REM *
130 REM *  DI ANDREA CAMPIONE
160 :
170 POKE36879,98
180 POKE36869,240
190 POKE36878,15
200 PRINT"[illegible]======FLIP!=========="
210 GOSUB750
220 PRINT"DEVI FAR SCOPPIARE I   PALLONI IN ALTO AL-"
230 PRINT"LO SCHERMO CON I DUE"
240 PRINT"OMINI ACROBATI:GUI-"
250 PRINT"DA IL TRAMPOLINO PER"
260 PRINT"ACCHIAPPARLI AL VOLO"
265 PRINT"USANDO I TASTI -M-N-"
270 PRINT"[illegible]PREMI UN TASTO..."
280 GETA$:IFA$=""THEN280
290 POKE36869,255
300 FORI=7168TO8185:POKEI,0:NEXT
310 P=0:Q=8174:W=8175
320 V=2:N=3:R=Q-22
330 FORI=7176TO7183:READA:POKEI,A:NEXT
340 FORI=7184TO7191:READB:POKEI,B:NEXT
350 FORI=7192TO7199:READC:POKEI,C:NEXT
360 FORI=7200TO7207:READD:POKEI,D:NEXT
370 FORI=7208TO7215:READE:POKEI,E:NEXT
380 FORI=7216TO7223:READF:POKEI,F:NEXT
390 POKEQ,V:POKEW,N:GOSUB630
400 IFQ=2THENPOKEQ,V:POKEW,N:POKER,0:R=W-22
410 IFW=5THENPOKEQ,V:POKEW,N:POKER,0:R=Q-22
420 IFPEEK(R-22)=1THENP=P+100:POKER-22,0:GOSUB680:GOTO490
430 IFR-22<7680THENGOTO490
440 H=INT(10*RND(1))
```

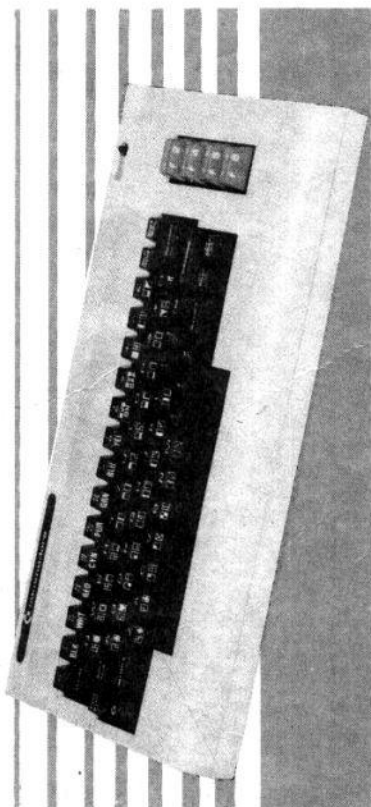
```
450 IFH>5THENGOSUB730:POKER,0:POKER-21,6:R=R-21
460 IFH<5THENGOSUB730:POKER,0:POKER-23,6:R=R-23
470 FORX=1TOL:NEXT
480 GOSUB560:GOTO420
490 IFPEEK(R+22)=3ORPEEK(R+22)=4THENGOTO800
500 IFPEEK(R+22)=2THENV=4:N=5:POKEW,N:POKEQ,V:POKER,0:GOSUB680:GOTO400
510 IFPEEK(R+22)=5THENN=3:V=2:POKEW,N:POKEQ,V:POKER,0:GOSUB680:GOTO400
520 IFR+22>8185THENGOTO800
530 POKER,0:POKER+22,6:R=R+22
540 FORX=1TOL:NEXT
550 GOSUB560:GOTO490
560 IFPEEK(197)=36THENGOTO590
570 IFPEEK(197)=28THENGOTO610
580 RETURN
590 IFW>8185THENPOKEQ,0:Q=8164:W=8165:POKEQ,V:POKEW,N:RETURN
600 POKEQ,0:POKEW,0:POKEQ+1,V:POKEW+1,N:Q=Q+1:W=W+1:RETURN
610 IFQ<8164THENPOKEQ,0:POKEW,0:Q=8184:W=8185:POKEQ,V:POKEW,N:RETURN
620 POKEQ,0:POKEW,0:POKEQ-1,V:POKEW-1,N:Q=Q-1:W=W-1:RETURN
630 PRINT"A A A A A A A A A A A"
640 PRINT" A A A A A A A A A A"
650 PRINT"A A A A A A A A A A A"
660 PRINT" A A A A A A A A A A"
670 RETURN
680 POKE36874,128
690 FORX=1TO10:NEXT:PRINT"1UP"P
700 POKE36874,0
710 FORX=1TO100:NEXT
720 RETURN
730 IFPEEK(R-21)=1ORPEEK(R-23)=1THENGOSUB940:RETURN
740 RETURN
750 PRINT"LIVELLO DIFFICOLTA'..."
760 PRINT"0=DIFFICILE 100=FACILE"
770 INPUTL
780 IFL>100THENPRINT"TROPPO!FIFONE!":GOTO770
790 RETURN
800 POKE36879,110
810 PRINT"PECCATO!!!!!!!!!!!!!!"
820 POKE36877,200
830 POKE 198,0:FORI=1TO1000:NEXT
840 POKE36877,0
850 PRINT"PREMI UN TASTO....."
860 GETA$:IFA$=""THEN860
870 RUN
880 DATA60,126,255,255,255,255,126,60
890 DATA8,4,2,1,0,0,0,0
900 DATA44,44,63,13,141,84,52,20
910 DATA52,52,252,176,177,42,44,40
920 DATA16,32,64,128,0,0,0,0
930 DATA90,90,126,24,24,36,66,66
940 P=P+100:PRINT"1UP"P:POKER,0:GOSUB680:RETURN

READY.
```

MODIFICA DEI COMANDI BASIC ORIGINALI

Questo articolo e i programmi in esso contenuti sono utili nel caso in cui si desideri modificare alcuni comandi Basic, poco usati nel Commodore 64, con altri rispondenti a particolari esigenze personali.

Si presuppone solo una piccola conoscenza del linguaggio macchina, cosicché con un po' di attenzione e buona volontà le seguenti righe saranno accessibili a tutti.

Per raggiungere il nostro scopo, è prima di tutto necessario capire come ogni normale comando è interpretato e fatto funzionare, dopodiché l'effettuazione delle modifiche diviene una naturale conseguenza. In R.O.M., da 41118 a 41373, c'è una tabella di 256 byte dove sono memorizzate le parole chiave, carattere per carattere, in codice ASCII (vedi figura 1). Da questa tabella mancano solo TI, TI$, ST, pi greco e GET#, gestiti in modo diverso e non influente ai nostri fini.

Ogni parola viene separata dalla seguente mediante l'aumento del codice dell'ultimo carattere di un numero fisso pari a 128; al RUN, in caso di programma (o alla pressione del RETURN in caso di comando diretto) l'interprete Basic compara ogni parola chiave con quelle inserite nella tabella mediante una ricerca sequenziale e, nello stesso tempo, tiene il conto delle parole lette, in modo da avere il numero d'ordine del comando.

Se la ricerca non sortisce alcun effetto, allora si ha il salto ad una routine che stampa sul video SYNTAX ERROR. In caso di esito positivo, invece, il numero d'ordine indica la posizione in un'altra tabella, dove risiede l'indirizzo, diminuito di una unità, della routine in linguaggio macchina relativo alla parola chiave.

Perciò, ricapitolando:

- c'è una tabella in ROM che contiene il codice ASCII, carattere per carattere, di ogni parola chiave;
- esiste un'altra tabella in ROM contenente gli indirizzi di inizio delle routine in linguaggio macchina relative a ciascun comando;
- in memoria sono conservate anche le routine in l.m. che sortiscono l'effetto desiderato dai vari comandi impartiti.

Per mettere in pratica i nostri intendimenti, dobbiamo tenere in debito conto tutte queste cose. Le nostre nuove parole chiave devono essere memorizzate, carattere per carattere, nella prima tabella; le relative routine in l.m. devono trovar posto in una zona di memoria libera e sicura da possibili sovrapposizioni; gli indirizzi iniziali di ciascuna nuova routine devono essere opportunamente posizionati nella seconda tabella.

Tutto questo chiaramente comporta delle alterazioni all'interprete Basic ma, poiché esso è residente in ROM, prima di continuare è necessario aggiungere qualcos'altro.

Il BASIC ufficiale

L'interprete Basic è posto tra le locazioni $AØØØ (40960) e $BFFF (49151) che, come abbiamo detto, costituiscono 8K di memoria a sola lettura (ROM). Esistono però, sotto la stessa, 8K di memoria RAM aggiuntiva, del tutto inusata, ed una porta (la locazione 1) che si occupa di selezionare un banco o l'altro di memoria. Quando tale porta contiene il valore 55 allora il sistema seleziona la ROM ed esclude la RAM sottostante, mentre col contenuto uguale a 54 si ha l'utilizzo della RAM con la conseguente esclusione della ROM.

Tenendo presente che, in condizioni normali (con la ROM attivata e la RAM esclusa) in questa zona qualsiasi tentativo di lettura riporta il contenuto della ROM ed ogni tentativo di scrittura modifica la RAM sottostante e non la ROM, possiamo risolvere il nostro problema usando un semplice ciclo che legge il contenuto dell'interprete e lo ricopia in RAM, dopodiché, ponendo nella locazione 1 il valore 54, si comunica al sistema di attivare la RAM al posto della ROM. Solo in questo modo il microprocessore non va in crisi in quanto continua a trovare al loro posto (che siano in RAM e non in ROM per lui non fa alcuna differenza) le varie routine che gli servono. Ora che abbiamo tutto in RAM, potremo effettuare quelle modifiche di cui sopra senza alcun problema.

In Basic, il ciclo che copia l'interprete da ROM in RAM è il seguente:

Forx = 40960 TO 49151 : POKE X, PEEK (X) : NEXT X

In linguaggio macchina l'esecuzione è più rapida (vedi figura 2).

Fatto questo, per inserire i nostri nuovi comandi dobbiamo comportarci esattamente come il sistema operativo, e precisamente:

- memorizzare in un'area sicura la routine in linguaggio macchina che eseguirà il nuovo comando desiderato;
- sostituire una parola chiave già presente nella tabella di Figura 1 con la nostra nuova parola facendo in modo che queste abbiano la stessa lunghezza;
- memorizzare a partire dalla locazione specificata nella colonna "1° indirizzo" di Figura 1, il valore ASCII di ogni carattere della nostra nuova parola chiave, facendo attenzione ad aumentare di 128 il valore dell'ultimo carattere;
- diminuire di una unità la locazione iniziale della routine precedentemente memorizzata ed inserire tale valore nel formato byte basso-byte alto a partire dalla locazione specificata nella colonna "2° indirizzo" di Figura 1 (per eventuali chiarimenti circa la dizione byte basso-byte alto, leggere l'articolo di Alessandro De Simone a pagina 43 e seguenti di COMMODORE COMPUTER CLUB n. 8);
- informare il microprocessore che si vuole utilizzare la RAM al posto della ROM con un: POKE 1,54.

N.	Par. chiave	1° Indirizzo	2° Indirizzo
Ø	END	41118	40972
1	FOR	41121	40974
2	NEXT	41124	40976
3	DATA	41128	40978
4	INPUT≠	41132	40980
5	INPUT	41138	40982
6	DIM	41143	40984
7	READ	41146	40986
8	LET	41150	40988
9	GOTO	41153	40990
10	RUN	41157	40992
11	IF	41160	40994
12	RESTORE	41162	40996
13	GOSUB	41169	40998
14	RETURN	41174	41000
15	REM	41180	41002
16	STOP	41183	41004
17	ON	41187	41006
18	WAIT	41189	41008
19	LOAD	41193	41010
20	SAVE	41197	41012
21	VERIFY	41201	41014
22	DEF	41207	41016
23	POKE	41210	41018
24	PRINT≠	41214	41020
25	PRINT	41220	41022
26	CONT	41225	41024
27	LIST	41229	41026
28	CLR	41233	41028
29	CMD	41236	41030
30	SYS	41239	41032
31	OPEN	41242	41034
32	CLOSE	41246	41036
33	GET	41251	41038
34	NEW	41254	41040

TAVOLA 1 delle parole chiave.

'1° indirizzo' indica la locazione in cui comincia la memorizzazione della perola chiave carattere per carattere in codice ASCII.

'2° indirizzo' indica la locazione dove risiede il byte basso della routine in l.m. che esegue il comando relativo alla parola chiave.

Sono state tralasciate tutte quelle parole chiave (per la maggior parte funzioni) la cui sostituzione presuppone altre conoscenze di base e non ottenibile mediante un semplice cambiamento di indirizzo.

Come esempio pratico, vediamo un nuovo comando che ci permetterà di cambiare il colore del bordo e dello sfondo dello schermo del C64 senza far uso di alcuna POKE.

Come prima cosa, trasferisco l'interprete Basic da ROM in RAM mediante la linea già citata o servendosi del programma di Figura 2.

```
FIGURA 2

,C000 A9 00
,C002 A8
,C003 85 FC
,C005 A9 A0
,C007 85 FD
,C009 B1 FC
,C00B 91 FC
,C00D C8
,C00E D0 F9
,C010 E6 FD
,C012 A5 FD
,C014 C9 C0
,C016 D0 F1
,C018 60
```

In seguito preparo la routine che deve effettuare il lavoro prima accennato e la memorizzo in un'area sicura (scelgo di cominciare a partire dalla locazione $Ø19), come in Figura 3.

Fatto questo, decido di sostituire la parola chiave LET, quasi mai usata, col mio

nuovo comando al quale, in considerazione del fatto che LET è di tre lettere, assegno il nome COL (abbreviazione di 'Colour').

Dalla colonna "1° indirizzo" di Figura 1, in corrispondenza a LET, trovo la locazione di inizio dove memorizzare i codici ASCII relativi alle lettere C O L (ricordarsi di aumentare l'ultimo valore di 128), ed eseguo;

POKE 41150,67 (67 = valore ASCII della lettera 'C')

POKE 41151,79 (79 = valore ASCII di 'O')

POKE 41152,204 (204 = valore ASCII di 'L' + 128).

Dalla colonna "2° indirizzo" di Figura 1, sempre in corrispondenza a LET, trovo la locazione da cui cominciare a memorizzare nel formato byte basso-byte alto l'inizio, diminuito di una unità, della routine in l.m. che esegue il nuovo comando COL ($Ø19-1=$ØØ18) ed effettuo:

POKE 40988,24 (24 = corrispondente decimale del numero esadecimale 18)

POKE 40989,192 (192 = corrispondente decimale del numero hex. CØ).

Infine, informo il microprocessore di voler selezionare la RAM al posto della ROM con: POKE 1,54.

```
FIGURA 3

,C019 20 40 C0
,C01C 20 FD AE
,C01F A5 15
,C021 D0 23
,C023 A5 14
,C025 C9 10
,C027 B0 1D
,C029 48
,C02A 20 40 C0
,C02D 68
,C02E AA
,C02F A5 15
,C031 D0 13
,C033 A4 14
,C035 C0 10
,C037 B0 0D
,C039 8E 20 D0
,C03C 8C 21 D0
,C03F 60
,C040 20 8A AD
,C043 4C F7 B7
,C046 4C 48 B2
```

A questo punto, ogni qualvolta l'interprete incontrerà il nuovo comando COL, la cui corretta sintassi è: COL valore1, valore2, dove valore1 e valore2 sono due numeri tra Ø e 15, colorerà il bordo dello schermo col colore corrispondente a valore1, e lo sfondo con quello corrispondente a valore2 (eventuali errori di sintassi o quantità genereranno gli appropriati messaggi).

Per tornare al Basic "normale" basterà premere contemporaneamente RUN/STOP e RESTORE oppure digitare in modo diretto: POKE 1,55 e premere il RETURN, mentre per rientrare nel nostro Basic personale basterà digitare in diretto: POKE 1,54 e premere il RETURN.

Per finire è stato riportato il listato del programma descritto per la gioia di tutti coloro che, non avendo voluto o potuto seguirmi erano solamente interessati a potenziare il Basic del loro computer con un nuovo comando.

Giovanni Verrelli

RUFFINI

La scuola è ormai incominciata, e con essa anche le lezioni di matematica. Se state frequentando la seconda liceo scientifico vi imbatterete per la prima volta nella risoluzione delle equazioni di grado superiore al secondo.

Mentre per la risoluzione delle equazioni di secondo grado bastava applicare una semplice formula, per le equazioni di grado superiore la ricerca delle radici non è così immediata. Gli unici modi per risolvere tali equazioni (ad esclusione di casi particolari come le biquadratiche) sono:

a) la scomposizione in fattori di primo e secondo grado.

b) La ricerca di una o più radici e l'abbassamento di grado dell'equazione stessa usando la regola di Ruffini

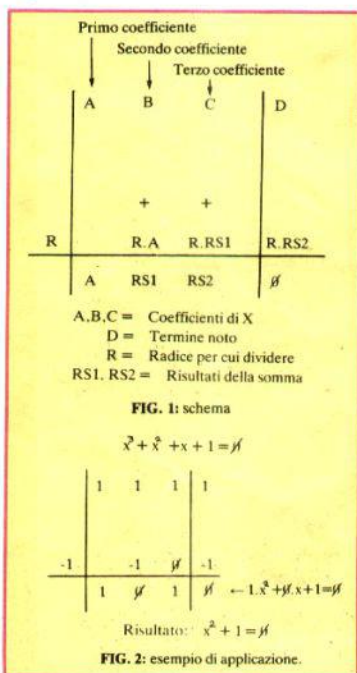
Come è noto questa si applica nella maniera seguente.

Si ricerca una radice sostituendo tutti i divisori positivi e negativi del termine noto, nella variabile x fino a quando si trova il numero che azzera l'equazione, che è appunto la radice cercata. Si scrivono poi i valori di tutti i coefficienti delle x ordinati secondo il grado descrescente delle x curando di inserire degli zeri per i termini mancanti all'interno del tipico schema a 'porta da rugby', come mostrato nell'esempio in figura 1.

Il termine noto (che sarà come è ovvio sempre presente) andrà inserito oltre la barra di destra. Si inserirà nello schema la radice trovata precedentemente, come mostrato in fig. 1, e si procederà nel modo seguente.

Si scrive il primo coefficiente sotto la riga orizzontale, lo si moltiplica per la radice e si scrive il risultato sopra la riga orizzontale incolonnandolo con il secondo coefficiente. Si somma ora algebricamente il secondo coefficiente con il risultato appena ottenuto e si scrive il risultato sotto la linea.

FIG. 1: schema

FIG. 2: esempio di applicazione.

Si continua ora ripetendo il procedimento per tutti i termini, moltiplicando il risultato appena ottenuto per la radice ecc. come mostrato ancora in figura 1.

E' evidente che il numero che andrà scritto sotto la riga orizzontale incolonnato con il termine noto dovrà sempre risultare uno zero, altrimenti significherà o che abbiamo sbagliato i conti o che il numero per cui abbiamo diviso l'equazione non era una radice della stessa.

A questo punto la divisione è effettuata. I numeri che troviamo scritti sotto la riga orizzontale sono i coefficienti ordinati in modo decrescente dell'equazione inferiore di un grado a quella data. Se l'equazione risultante dovesse essere ancora di grado superiore al secondo o non risolvibile per via elementare si ripeterà il procedimento descritto fino ad ottenere una equazione immediatamente risolvibile.

Vediamo ora un semplice esempio per chiarire il tutto e passiamo poi alla descrizione del programma.

x↑3 + x↑2 + x +1=0 è una semplice equazione di terzo grado. Proviamo a ridurla ad una equazione di secondo grado usando la regola di Ruffini.

Per prima cosa analizziamo i divisori del termine noto. Interi sono solo +1 e -1.

Iniziamo a provare con questi due. Sostituendo +1 al posto delle X otteniamo +4, quindi +1 non è il numero che cerchiamo.

Sostituendo -1 otteniamo 0, avendo conferma che -1 è una delle tre radici dell'equazione. Applichiamo ora la regola di Ruffini come mostrato in fig. 2 ed otteniamo x↑2 +1 = 0 che è appunto l'equazione di secondo grado risultante dalla divisione.

Vediamo ora come gira il programma.

Dato il RUN vi sarà chiesto il grado dell'equazione da dividere, e poi i coefficienti delle x in ordine decrescente di grado. L'equazione potrà essere al massimo di quarto grado, altrimenti lo schema non sarà simmetrico.

Una volta inseriti tutti i coefficienti, si potranno verificare due casi:

1/ dopo qualche secondo apparirà lo schema completamente risolto;

2/ dopo qualche secondo apparirà la scritta DIVISORE?

Nel primo caso significa che il computer ha trovato automaticamente una radice dell'equazione e che ha quindi già risolto tutto.

Nel secondo caso il computer ci comunica che non ha trovato divisori e chiede a

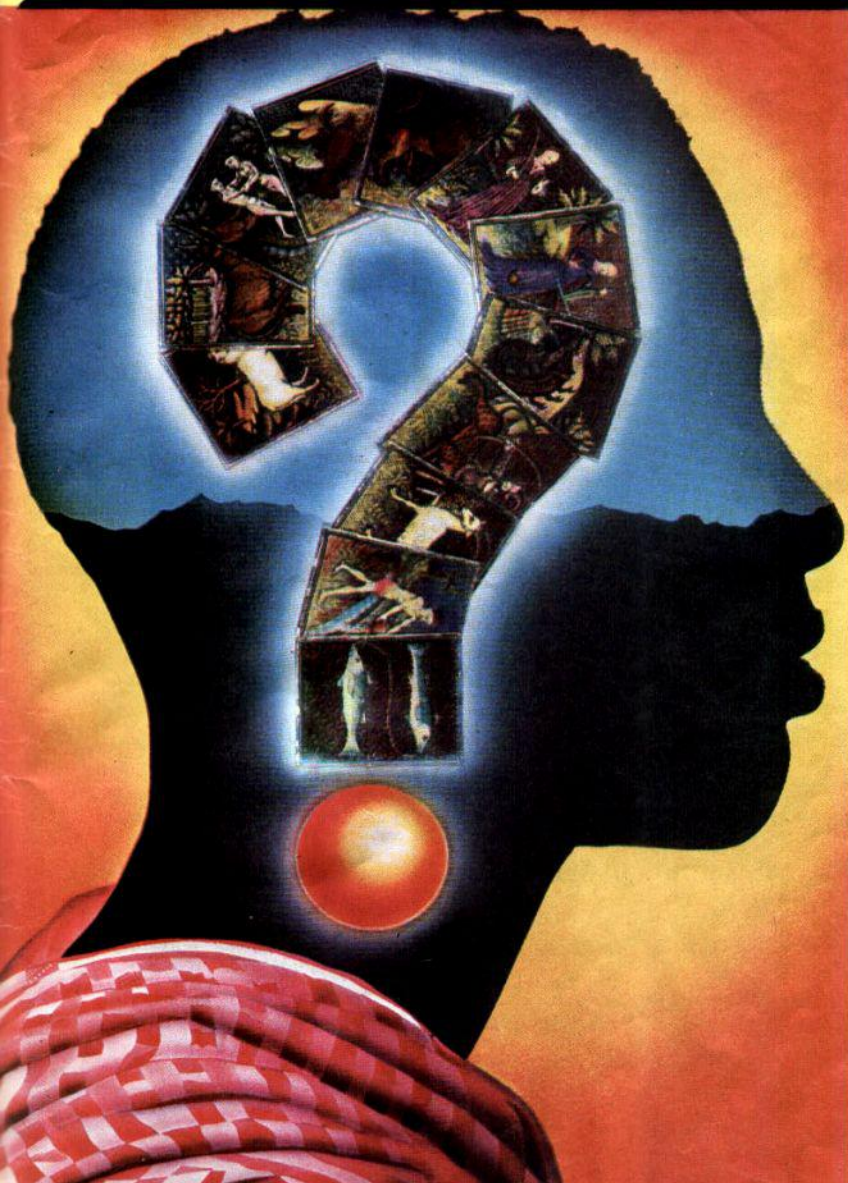
noi di cercarne uno. Una volta che lo abbiamo trovato basta inserirlo e il computer eseguirà tutti i calcoli per noi. Converrà anche controllare che il divisore non sia +/-3, in quanto il computer, pur controllando tutti gli interi "e mezzi" dei divisori del termine noto, per le inevitabili imprecisioni di calcolo, potrebbe non trovare il tre (provate a far eseguire PRINT 3↑4 e capirete perchè).

Con il tasto F3 effettuare un'ulteriore divisione del risultato ottenuto, nel caso che questo sia una equazione di grado ancora elevato.

N.B. Se avete digitato correttamente il programma, provate ad inserire i dati relativi alla seguente divisione:
$(2x^4-5x^3+6x^2-7x+1):(x-2)$

Il risultato dovrebbe essere:
$2x^3x^2+4x+1$ Resto = 3

Andrea & Alberto Boriani

```
100 REM *** RUFFINI INTERPRETATO DAI BORIANI ***
110 :
120 PRINT"┌──────────┐"
130 PRINT" RUFFINI |"
140 PRINT"└──────────"
150 IFX>=1THENPRINT"":A=X:GOTO190
160 CLR
170 INPUT"GRADO DELLA EQUAZIONE";A
180 DIMM(A+1):X=A
190 PRINT" ";TAB(6+6*A);"| "
200 PRINT" ";TAB(6+6*A);"| "
210 PRINT" ";TAB(6+6*A);"| "
220 PRINT" ";TAB(6+6*A);"| "
230 PRINT" ";TAB(6+6*A);"| "
240 PRINT"───+";:FORS=1TOA*6:PRINT"-";:NEXT:PRINT"+───"
250 PRINT" ";TAB(6+6*A);"| "
260 PRINT" ";TAB(6+6*A);"| "
270 IFP=1THENPRINT"":GOTO310
280 PRINT"COEFFICENTE DI X↑"A;:INPUT M(A)
290 A=A-1
300 IFA>=0THENPRINT"                                       ":GOTO280
310 PRINT"";
320 FORD=XTO0STEP-1:PRINTTAB(7+6*(X-D));M(D);:NEXT
330 DI=1
340 IFM(0)<0THENDI=-1
350 FORC=0TOM(0)STEPDI/2:Y=0
360 FORS=0TOX:Y=Y+C↑S*M(S):NEXTS
370 IFY=0THEN440
380 NEXTC
390 FORC=0TO-M(0)STEP-DI/2:Y=0
400 FORS=0TOX:Y=Y+C↑S*M(S):NEXTS
410 IFY=0THEN440
420 NEXTC
430 INPUT"DIVISORE";C
440 PRINT"";C;
450 F=0:FORD=XTO0STEP-1
460 M(D)=M(D)+F:F=M(D)*C
470 IFD<>0THENPRINTTAB(13+6*(X-D));F;
480 PRINT""
490 PRINTTAB(7+6*(X-D));M(D);"";:NEXT
500 GETF$:IFF$=""THEN500
510 IFX>=1THENFORI=1TOX:M(I-1)=M(I):NEXT:X=X-1:P=1:GOTO120

READY.
```

OFFRONO LA PROPRIA CONSULENZA

Pomeriggio solo per Foggia o provincia. **Leonardo De Palma** *Via Francesco Marinaccio, 7 cap 71100 Foggia Tel. 0881/79700* (pomeriggio).

Dimostratore e/o installatore di personal computer Commodore, anche a tempo pieno, zona Viterbo o Roma. **Andrea Di Palermo** *via Silvio Pellico, 14/A cap 01030 Vasanello (VT).*

Dimostratore offresi. **Giovanni Tomiolo** *Via Ca' Rotte, 13 cap 37053 Cerea Tel. 0442/82747* (ore 12-13 oppure 18-20).

Diciottenne perito elettrotecnico offresi per qualsiasi attività nel campo dei personal computer. **Francesco De Colle** *P.le Capo Linaro, 11/23 cap 00053 Civitavecchia (Roma) Tel. 0766/34171 oppure 40171* (ore pasti).

Consulenze e programmi su richiesta per Commodore 64. Tel. **Claudio** *02/680482* (ore 20,30-21,30) *Milano.*

Esperto Basic Commodore, studente programmazione, diplomato operatore con esperienza su sistema TI 990 S.O.DX10, offresi come dimostratore disponendo di ore libere al mattino. **Cristiano Monti** *via Marcello Moretti, 6 cap 20132 Milano Tel. 02/2856914* (ore 9-13).

Studente appassionato offresi come dimostratore presso punti di vendita di personal computer zona Nord Milano. **Roberto Quaglia** *via Martinazzoli, 2 cap 20161 Milano Tel. 02/6462130.*

Appassionato conoscitore del Commodore 64 disponendo tempo libero al pomeriggio offresi come dimostratore solo in zona. **Gioacchino Di Fazio** *via Cap. Ippolito, 62 cap 92024 Canicattì (AG) Tel. 851363/109221* (ore 14-16).

Studente di informatica offresi. **Luigi Monaco** *Via Largo Adua, 28 Bari.*

Studente secondo anno ingegneria informatica offresi come consulente hardware software su CBM 64. **Marco** *Tel. 039/460853 Lissone (MI).*

CONSULENZE CERCANO

Cerco consulenza tecnica e software per Commodore serie 8000 CBM 8096SK. **Michele Giustino** *Via Posillipo, 85 cap 80100 Napoli Tel. 081/7697205 oppure 7696025.*

Cerco consulenza per trasformazione Linguaggio Macchina per Olivetti M/20 programma personale per uso proprio. **Ditta Centro Sistemistico Cinà** *Via Aquileia, 68 Palermo Tel. 206684.*

Cerco urgentemente programmatore per realizzare programmi di Meccanica (Resistenze dei materiali) su Commodore 64. **Franco e Maurizio Bruno** *via Giorgio Bratti, 100 cap 47023 Cesena Tel. 0547/23810* (ore 18.30-20).

Scambiano programmi per Vic 20

Giuseppe Mieli Via Tica, 167 cap 96100 Sirasuca Tel. 33421

Alvaro Ceccarini Via Di Vittorio, 10 cap 58022 Follonica (GR) Tel. 0566/43248 (ore serali feriali)

Antonio Raineri Via Mazzini, 1 cap 20077 Melegnano (MI) Tel. 02/9833788 (ore 12.30-14 dal lunedì al sabato)

Federico Novelli Via Cavour, 15 cap 21013 Gallarate (VA)

Vendono hardware per Vic 20

Vic 20, espansioni, cartucce, manuali, libri, programmi. **Patrick** Tel. 02/6570068 solo zona Milano.

Vic 20, espansione 16K, registratore Commodore. **Claudio Morgia** Via dei Zeno, 31 cap 00176 Roma Tel. 2772314 (ore 11-18)

Vic 20, espansione 16K, Superexpander, libri in italiano e inglese. **Andrea Espositi** Via Brescia, 16 cap 00196 Roma Tel. 8458011 (ore pasti)

Vic 20, registratore, manuali e software vario. **Roberto Covanti** Fabriano Tel. 0732/3915

Vic 20, cartuccia, software vario. Massa Carrara. Tel. 0585/57972 **(Luca)**

Vic 20, manuali, libri, software vario. **Massimo Mazzoni** Via Varsavia, 8 cap 50065 Pontassieve (FI) Tel. 055/8302440 (dopo le 19.30)

Registratore Commodore. **Gatti Don Ferdinando** Via S. Giovanni Bosco, 1 cap 20084 Lacchiarella Tel. 9008121 (ore 12-13 e 19-20)

Comprano periferiche Commodore

Disk drive max 350000 Roberto Tel. 070/961121

Disk drive possibilmente doppio compatibile con PET 3032. **Giuseppe Nipoti** Via Malvasia, 10 cap 40131 Bologna Tel.

051/552591 (ore pasti)
Disk drive per C-64 **Gatti Don Ferdinando** Via S. Giovanni Bosco, 1 cap 20084 Lacchiarella Tel. 9008121

PICCOLI ANNUNCI SOFTWARE

Avvertenze Onde evitare speculazioni commerciali, da questo numero vengono pubblicati solo i nominativi dei lettori con la sola dizione SCAMBIANO PROGRAMMI e con la suddivisione per computer.
È pertanto inutile inviare elenchi interminabili di titoli che non verranno presi in considerazione.

Scambiano programmi per Commodore 64

Giovanni Addabbo Via Monte Vettore, 46 cap 60131 Ancona
Mauro Iannucci Via E. Fermi, 49 cap 00146 Roma
Silvio Gatta Via L. Calori, 1 cap 40122 Bologna Tel. 556218 (ore pasti)
Maurizio Scibilia Via Alighieri, 30 cap 53040 Rapolano Terme (SI) Tel. 0577/-724205
Nico Tranquilli Via Ruffini, 13 cap 62012 Civitanova M. Tel. 0733/770332
Claudio Veronese Via M. Bensi, 29 cap 15100 Alessandria Tel. 0131/344591
Angelo Orlandi Via delle Albizze, 40 cap 00172 Roma Tel. 288368
Firmino Cigalotto Via Laipacco, 283 cap 33100 Udine Tel. 0432/284563
F.lli Cervi lotto Z is.M cap 80144 Via S. Claudio Minopoli **Napoli** Tel. 081/-7011322 (ore 18-22)
Flavio Giuliano Via Costantino, 22 cap 00145 Roma Tel. 5127007 (ore 14-15 oppure 20.30-22.00)
Graziano Baleani Via De Gasperi, 42 cap 60020 Agugliano (AN) Tel. 907093 (ore 15-18)
Paolo Cangianelli Via Oslavia, 43 cap 00195 Roma Tel. 06/385173 (ore pasti)
Massimo Leoncini Casella Postale, 85 cap 16035 Rapallo (GE)
Luca Dell'Oro Via dei Lillà, 53 cap 20089 Rozzano (MI) Tel. 02/8251907 (ore 13-16 oppure 19.30-21)

CERCASI

La redazione di Commodore Computer Club vuole potenziarsi e ricerca collaboratori part-time e insegnanti di discipline scientifiche e tecniche preferibilmente residenti nell'area di Milano

Ai collaboratori che stiamo ricercando verrà richiesto di collaborare alle varie iniziative della casa editrice con articoli, libri, raccolte di programmi e l'italianizzazione di software, di cui abbiamo i diritti d'autore, orientati alla didattica per le scuole medie e superiori.
I prescelti, pertanto dovranno possedere un sistema completo di Vic 20 oppure Commodore 64 e sapere programmare sia in basic che in linguaggio macchina. La conoscenza dell'inglese e di altri personal computer è un titolo preferenziale.

Compensi

Tutti i lavori svolti su incarico della redazione verranno sempre compensati in base ai miglior standard di mercato.

Primo contatto

Per incontrarci telefonate allo 02/8467348 chiedendo della signorina Piera

LISTINO PREZZI COMMODORE

COMPUTER

PRODOTTO	PREZZO (IVA ESCL.)
VIC 20	
HOME COMPUTER Memoria base 5 Kbytes, 255 combinazioni di colori, 3 voci, BASIC residente, si collega direttamente a qualsiasi televisore domestico.	199.000
COMMODORE 16	
HOME COMPUTER Memoria a 16 Kbytes RAM standard, 12 Kbytes RAM accessibili per programmi in BASIC. BASIC 3.5 residente. 121 colori (15 di base con 8 gradazioni + il nero). Collegabile a un comune televisore.	245.000
COMMODORE 64	
CPU 64K RAM Computer con alta risoluzione grafica, 256 combinazioni di colori, sintetizzatori di suono. Possibilità di creare eccezionali figure tridimensionali dinamiche (sprites). Tastiera grafica. Dimensione dello schermo 40 colonne per 25 righe. Compatibile con tutte le periferiche Commodore. Collegabile a un comune televisore.	625.000
C 64 EXECUTIVE Il solo e fantastico computer portatile della seconda generazione. Monitor a colori di 5" ad alta risoluzione grafica e Floppy Disk Drive da 170 Kbytes incorporati! Tastiera grafica separabile, suono professionale e completa compatibilità con i programmi e le periferiche del Commodore 64 ne fanno un computer dai mille usi. Può inoltre utilizzare tutte le cartucce C 64 e può essere collegato al monitor 1702.	2.350.000
COMMODORE PLUS 4	
PERSONAL COMPUTER Memoria 64 Kbytes RAM, 60 bytesK RAM accessibili per programmi in BASIC. BASIC 3.5 residente. 121 colori (16 di base con 8 gradazioni). 4 programmi software residenti su ROM. Collegabile a un comune televisore.	975.000
COMMODORE 8000	
PERSONAL COMPUTER CPU 128 Kbytes RAM, 24Kbytes ROM, Basic 4.0 residente, video orientabile e basculante 80 colonne per 25 righe, tastiera commerciale separata. Software compatibile con CBM 8032 e CBM 8096.	1.995.000
COMMODORE 700	
PERSONAL COMPUTER CPU 128 Kbytes RAM espandibili internamente a 256 Kbytes ed esternamente a 960K. Video orientabile e basculante, 80 colonne per 25 righe. Compatibile con tutte le periferiche Commodore delle serie professionale.	2.850.000
PERSONAL COMPUTER CPU 256 Kbytes RAM espandibili esternamente a 960 Kbytes. Video orientabile basculante, 80 colonne per 25 righe. Compatibile con tutte le periferiche Commodore delle serie professionale.	3.250.000

PERIFERICHE

PRODOTTO	COMPUTER COLLEGABILI	PREZZO (IVA ESCL.)
MONITOR		
MONITOR MONOCROMATICO A fosfori verdi 12"	VIC 20, C 16, C 64, SX 64, PLUS 4	285.000
MONITOR A COLORI Ad alta risoluzione, 14", con audio.	VIC 20, C 16, C 64, SX 64, PLUS 4	690.000
REGISTRATORI		
REGISTRATORE DEDICATO Per memorizzare facilmente programmi e dati su normali cassette magnetiche.	VIC 20, C 64,	120.000
REGISTRATORE DEDICATO Per memorizzare facilmente programmi e dati su normali cassette magnetiche.	C 16, PLUS 4	120.000
MEMORIE DI MASSA		
FLOPPY DISK DRIVE Unità di memoria di massa, drive singolo, capacità 170 Kbytes in linea.	VIC 20, C 16, C 64, SX 64, PLUS 4	630.000
FLOPPY DISK DRIVE Unità di memoria di massa, drive doppio, capacità 1 Mbytes in linea.	C 8032SK, C 8096, C 8296, C 710, C 720, C 610, C 620	2.375.000
FLOPPY DISK DRIVE Unità di memoria di massa, drive singolo, capacità 1 Mbytes in linea.	C 8032SK, C 8096, C 8296, C 710, C 720, C 610, C 620	1.245.000

Riservato agli ingegneri

Il miglior software tecnico su elaboratori CBM - Commodore
Ora anche disponibile su Commodore 64

"STR/FOND"
L'ormai famoso programma per il calcolo delle strutture intelaiate piane in c.a., in zona sismica, che sviluppa e disegna anche le carpenterie delle armature, ora agganciato al programma "FONDAZIONI" che risolve l'intero graticcio di fondazione proponendo una carpenteria sofisticata ed ottimizzata.
L. 1.200.000 (versione giugno '84 - nostra esclusiva)

"MURI DI SOSTEGNO"
A gravità, a mensola ed a contrafforti, anche in zona sismica, secondo il D.M. del 21.1.81
L. 300.000

"PENDII"
Analizza la stabilità di un pendio o di un fronte di scavo sotto diverse condizioni e la verifica relativa viene condotta in termini di tensioni effettive: la stima dei fattori di sicurezza viene effettuata secondo i metodi di Fellenius, Bishop e Janbu.
L. 300.000

"PALI"
Esegue la progettazione delle armature e le verifiche di sicurezza per fondazioni su pali (plinti o platee). E' previsto anche il plinto senza pali, con schema, in tal caso, di suolo elastico alla Winkler.
L. 700.000

"A.S.C.I." (Analisi Strutture per Coperture Industriali)
Calcolo di travature reticolari formanti strutture (in acciaio) per coperture industriali. Verifiche di resistenza e di instabilità laterale per ogni asta. Analisi delle sollecitazioni indotte nei pilastri sotto carichi trasmessi dalle travature e azioni sismiche.
L. 600.000

"COMPUTI METRICI"
Analisi ed elenco prezzi. Metodo veloce e complementare automatizzato per il computo e la stima dei lavori.
L. 500.000

"REVISIONE PREZZI"
Secondo le disposizioni di legge vigenti. Praticità ed automazione consentono di eseguire velocemente revisioni di prezzi anche per lunghi periodi.
L. 300.000

"ORARIO SCOLASTICO"
Nuovo ed esclusivo "pacchetto" che consente la stesura dell'orario scolastico per ogni tipo di scuola. Indispensabile per la velocità e l'imparzialità dell'elaborazione.
L. 700.000

Richiedeteci documentazione e output dei programmi di vostro interesse.
Resterete sbalorditi dalla versatilità e dalla completezza del nostro software

SIRANGELO COMPUTER Srl
Via Parisio, 25 - Cosenza 0984 - 75741

INTERFACCIA REGISTRATORI A CASSETTE PER VIC 20 E COMMODORE 64

Adatta tutti i normali registratori a cassetta al tuo computer. Ti permette di duplicare i programmi da un altro normale registratore. Con sole **34.000** lire I.V.A. e spedizione compresa potrai ricevere direttamente a casa tua questa indispensabile interfaccia, inviando il buono di ordinazione accuratamente compilato.

BUONO DI ORDINAZIONE

Inviatemi N. ______ interfacce cassette

Sig. ______________________________

Via ______________________________ N. ______

cap __________ Città ______________________ (______)

R.C.P. ELETTRONICA SRL

Via Don Pasquino Borghi, 13
42017 NOVELLARA (REGGIO E.)
Tel. 0522/661471

STAMPANTI		
STAMPANTE PLOTTER A COLORI 80 caratteri per linea, 4 colori, alta risoluzione di 0,2 mm per passo.	VIC 20, C 16, C 64, SX 64, PLUS 4	375.000
UNITÀ STAMPANTE Stampa velocemente su carta normale quanto appare sul video: programmi, lettere, dati, grafici.	VIC 20, C 16, C 64, SX 64, PLUS 4	515.000
FLOPPY DISK DRIVE Unità di memoria di massa, drive doppio, doppia faccia, doppia densità, capacità 2 Mbytes in linea.	C 8032SK, C 8096, C 8296, C 710, C 720, C 610, C 620	2.600.000
FLOPPY DISK DRIVE Unità di memoria di massa, drive doppio, doppia faccia, doppia densità, capacità 2 Mbytes in linea.	C 8032SK, C 8096, C 8296, C 710, C 720, C 610, C 620	2.600.000
STAMPANTE Bidirezionale, 80 CPS, 80 colonne, carta in modulo singolo e/o continuo.	VIC 20, C 16, C 64, SX 64, PLUS 4	645.000
STAMPANTE A COLORI 7 colori, stampa a matrice di punti, 38 CPS, 80 colonne.	VIC 20, C 16, C 64, SX 64, PLUS 4	995.000
STAMPANTE A MARGHERITA Bidirezionale, 17 CPS, trascinamento a frizione, stampa fino a 165 caratteri per linea.	VIC 20, C 16, C 64, SX 64, PLUS 4	1.195.000
STAMPANTE Bidirezionale ad aghi, 60 CPS, 80 colonne.	C 8032SK, C 8096, C 8296, C 710, C 720, C 610, C 620	695.000
STAMPANTE Stampante ad aghi, 150 CPS, 132 colonne, bidirezionale, trascinamento a trattore e frizione.	C 8032SK, C 8096, C 8296, C 710, C 720, C 610, C 620	1.275.000
STAMPANTE Stampante a margherita, 40 CPS, 136 colonne passo pica, 163 colonne passo élite, bidirezionale, utilizzabile anche con carta da bollo, trascinamento a trattore e frizione.	C 8032SK, C 8096, C 8296, C 710, C 720, C 610, C 620	3.250.000

ACCESSORI

PRODOTTO	COMPUTER COLLEGABILI	PREZZO (IVA ESCL.)
INTERFACCIA IEEE 488 Consente il collegamento di tutte le periferiche con interfaccia IEEE 488.	C 16, C 64, SX 64, PLUS 4	175.000
COMANDO PER GIOCHI (Joystick) Permette di muoversi in tutte le direzioni, di iniziare i vari giochi di movimento e sparare.	VIC 20, C 64, SX 64	13.500
COMANDO A MANOPOLA PER GIOCHI (Paddle) Adatto per i giochi a 2 persone, esegue movimenti in orizzontale e verticale.	VIC 20, C 64, SX 64	22.500
NUOVO SISTEMA OPERATIVO Per 8096SK o per 8032SK con B. Può gestire fino a 16 programmi residenti simultaneamente in memoria. Dà a disposizione 26K per le variabili e 53K per i programmi. Potenzia inoltre il Basic con altri comandi.	C 8032SK, C 8096, C 8296	95.000
64K RAM Scheda di ampliamento memoria per 8032 e nuovo sistema operativo "PM 96".	C 8032SK	575.000
CP/MAKER Incrementa la memoria interna di 64K RAM e permette l'uso di tutti i programmi CP/M. 8 bit disponibili. Compatibile con la serie 3000/4000/8000.	C 8032SK	1.450.000
SCHEDA AD ALTA RISOLUZIONE GRAFICA Compatibile ai sistemi della serie 8000.	C 8032SK, C 8096, C 8296	720.000
CAVO PET/IEEE	C 8032SK, C 8096, C 8296, C 710, C 720, C 610, C 620	85.000
CAVO IEEE/IEEE	C 8032SK, C 8096, C 8296, C 710, C 720, C 610, C 620	95.000
ACCOPPIATORE ACUSTICO 300 baud.	C 8032SK, C 8096, C 8296, C 710, C 720, C 610, C 620	595.000

INVIARE TUTTA LA PAGINA ANCHE SE SI UTILIZZA UNA SOLA SCHEDA

Nome ______________________ Cognome ______________________

Via ______________________ Nº __________ CAP. __________ Città ______________________

Telefono ______________________ Orario ______________________

Registrate il mio abbonamento annuale a Commodore Computer Club.

☐ Ho versato oggi stesso il canone di L. 22.000 a mezzo c/c postale n° 31532203 intestato a:
Commodore Computer Club - V.le Famagosta, 75 - 20142 Milano

☐ Ho inviato oggi stesso assegno bancario n...
per l'importo di L. 22.000 intestato a Commodore Computer Club.
Si prega di scrivere il proprio nome e l'indirizzo completo in modo chiaro e leggibile. Inviare la fotocopia del bollettino di c/c postale.

Consideriamo che i numeri 1 e 2 sono esauriti, vogliate inviarmi i numeri arretrati .... al prezzo di L. 5.000 cadauno per richieste fino a 4 numeri, o di L. 4.000 cadauno per richieste oltre i 4 numeri arretrati, e perciò per un totale di L............... Sono a conoscenza che i fascioli suddetti non saranno inviati in contrassegno e, pertanto, ho provveduto oggi stesso a versare il canone di L................. a mezzo c/c postale n. 31532203 intestato a:
Commodore Computer Club - V.le Famagosta, 75 - 20142 Milano

STATISTICA

Non posseggo un computer ☐
Posseggo un C64 .... si ☐ ... no ☐
Posseggo un VIC 20 .... si ☐ ... no ☐
Posseggo un Commodore Plus 14 .... si ☐ ... no ☐
Posseggo un Commodore Plus 16 .... si ☐ ... no ☐
Posseggo un registratore dedicato .... si ☐ ... no ☐
Posseggo un drive 1541 .... si ☐ ... no ☐
Posseggo una stampante .... si ☐ ... no ☐
Posseggo un monitor .... si ☐ ... no ☐

COLLABORAZIONE

A titolo di prova vi invio un articolo e la cassetta........................disco....................................
col programma che intendo proporre per la pubblicazione di cui garantisco l'originalità.

DOMANDA/RISPOSTA ....

....

....

....

....

....

RICHIESTA ARGOMENTI

Mi farebbe piacere che Commodore Computer Club parlasse più spesso dei seguenti argomenti:

1/ ....

2/ ....

3/ ....

4/ ....

GIUDIZIO SUI PROGRAMMI DI QUESTO NUMERO

Ho assegnato un voto da 0 a 10 ai programmi che indico di seguito:

A/ .... Voto ....

B/ .... Voto ....

C/ .... Voto ....

D/ .... Voto ....

PICCOLI ANNUNCI..

....

....

....

....

....

CERCO/OFFRO CONSULENZA..

....

....

....

....

....

INVIARE IN BUSTA CHIUSA E AFFRANCANDO SECONDO LE TARIFFE VIGENTI A:

COMMODORE COMPUTER CLUB

V.le Famagosta, 75
20142 Milano

INVIARE TUTTA LA PAGINA ANCHE SE SI UTILIZZA UNA SOLA SCHEDA

Nome ______ **Cognome** ______

Via ______ **N°** ______ **CAP.** ______ **Città** ______

Telefono ______ **Orario** ______

Commodore Club
SPECIAL
Suppl. al n. 3
Lire 10.000
Fantastico!
Ecco la cassetta che fa parlare il tuo 64
La voce!
e fai parlare il tuo computer